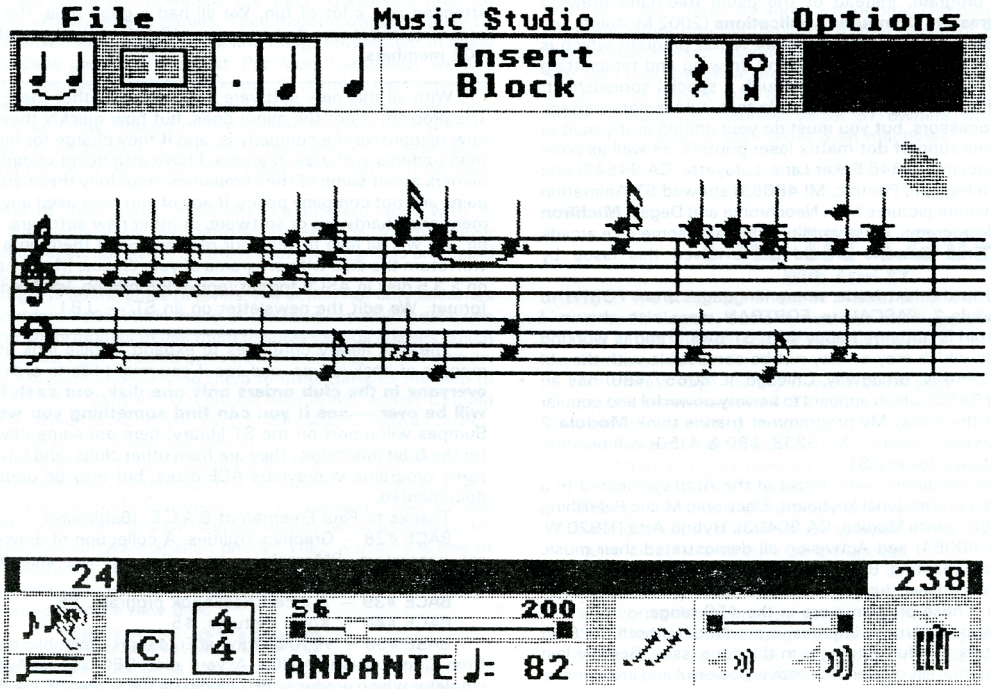
ATARI COMPUTER ENTHUSIASTS

3662 Vine Maple Dr. Eugene OR 97405

JUNE, 1986

Mike Dunn, Jim Bumpas, Larry Gold, co-editors



News and Reviews

by Mike Dunn, Co-Editor

Last month, I listed some of the things I saw at Comdex in Atlanta. This month, I will expand my impressions. I arrived at the fabulous Atlanta airport, with its computer-generated voices telling you where to go on the comperized automatic trains which run all over. On to the Marriott-Marquis, with another computer voiced vehicle — this time a talking elevator. Just the thing to get a person in the mood for the huge Comdex show. There were 1000 booths and 45,000 people at the show!

The Atari booth consisted of a section for the Atari and many smaller booths for 3rd party vendors — about 30 or so. The emphasis was on business and home productivity for the Atari ST, but still much for us 8-bit fans. Very little in game software was shown, a big contrast with the Amiga booth, which seemed to be mostly games, music and art. Commordore annouced the first data-base for the Amiga at the show; there are many available for the ST. I do not yet have an ST, but am impressed with the great enthusiasm of the vendors, the powerful and easy to use software, the the very low prices for super programs. Many wonderful word-processors, full of features for $40 to $100. Especially noteworthy was the **Thunder** program from Batteries Included (30 Mural Richmond Hill, Ontario, Canada L4B 1B5), a real-time quick and powerful spelling checker which does everything and more, for only $40. Many new products from this Canadian Company who brought us **PaperClip** and **Degas** were shown. New, expanded versions of PaperClip with a spelling checker, in both 8-bit and 16-bit forms, as well as Degas, the Isgur Portfolio ($200), and many others. Remember, this was one of the few companies releasing Atari software when most had given up — they deserve you support.

Many companies were showing various word processors, spelling checkers, file managers and data bases, etc. — last month's issue lists many of them. Atari will be distributing dbMAN, a dBase III-alike program from Versasoft of San Jose for $150.

There were several very impressive graphic products looking like they will be great fun to have. **Hippo** (985 University #12, Los Gatos, CA 95030) and **Print-Technik** (2, D-8000, Munchen 40, W. Germany) showed video-digitizers ($130 to $250) which were very nice. **Xlent** (POB 5228, Springfield, VA 22150) had new ST versions of their popular Typesetter and Page Designer 8-bit programs, as well as some new 8-bit programs including a wordprocessor and many other products. **PrintMaster** (Unison World, 2150 Shattuck, #902, Berkeley, CA 94704) showed a nice ST "Printshop" type program. **Migraph** (720 S. 333rd St., Federal Way, WA 98003) had EasyDraw, a very powerful object-oriented drawing program, instead of the usual free-hand drawing program, and **Progressive Computer Applications** (2002 McAuliffe Dr., Rockville, MD 20851), showed their **Graphic Artist** program which is a combination of a CAD, graphics, free-hand drawing and typesetting program for $500. Everything is done through a special spreadsheet, and I understand from talking to local people that it does not interface with other wordprocessors, but you must do your editing in the built-in spread sheet. It does support dot-matrix laser printers, as well as color plotters. Avila Associates (3646 Baker Lane, Lafayette, CA 94549) and Michtron (576 S. Telegraph, Pontiac, MI 48053) showed ST Animation programs which animate pictures from Neochrome and Degas. **MichTron** has many other ST programs, mostly utilities, but also some nice arcade games — their **Time Bandits** should be reviewed this issue by Jim Bumpas.

As mentioned in the last issue, many languages from FORTH to COBAL, C to Modula-2, PASCAL to FORTRAN were also shown. I understand that Atari is not very happy with ST BASIC and is working on a major upgrade which may or may not be compatible with the old one. **Softworks** (2944 N. Broadway, Chicago, IL 60657 $80) has an Alpha-Micro based BASIC which appears to be very powerful and popular with the people in the know. My programmer friends think **Modula-2** by TDI (10410 Markison, Dallas, TX 75238, $80 & $150) will become the language of choice for the ST.

The most dramatic demos were those of the Atari connected to a MIDI keyboard and a professional keyboard. Electronic Music Publishing (2210 Wilshire #488. Santa Monica, CA 90403), Hybrid Arts (11920 W. Olympic, L.A., CA 90064) and Activision all demostrated their music programs. At our last meeting, Bob Cook played the Activision program hooked to an inexpensive (under $300) CASIO synthesizer with impressive results. These programs are in the $50 range.

ANTIC! magazine had many impressive products for both the 8-bit and 16-bit computers; the full catalog is in the June issue. See the last issue of ACE for a partial list of their innovative, powerful and inexpensive line. It is definitely worth buying this issue if you don't already have a subscription. (POB 1919 Marion, OH 43306 ANTIC! $28year, STart (the ST quarterly including a 3½ disk of programs, and ANTIC! for $60 year).

The **MindMine** (13256 NE 20th #4, Bellevue, WA 98005) was one of the outstanding booths because of the wonderful personality of its owner, Lee Rahfeldt and his 10 year old daughter, who has the distiction of being the youngest vendor ever at COMDEX. Lee is a delightful man who makes a 1 Meg ST upgrade, a real time clock module and a custom talking doll for your ST — you send them a photo, and they try to duplicate it. He is also selling the new **MT-FORTH** by an old friend of ACE and FORTH, Pat Mullarky ($50), originator of the SPACE and APX FORTH'S, as well as public domain versions of the 8-bit FORTHs. Paul Heckel, owner of QuickView (146 Main St, #404, Los Altos, CA 94022) showed his original program **ZoomRacks** $80, the innovative electronic rack and his new book "The Elements of Friendly Software Design" published, by all people, Warner Books. Zoomracks is the type of program all ST owners should have, I understand. [We still have one copy remaining from our group purchase, if you want it for $48 — J.B.]

Atari was also there, remember, taking up the other half of the booth. They showed their new 20-meg Hard Disk for $700, and most of the other vendors were using it. They also had their MS-DOS box with 512K working; supposed to be ready this summer for $200-$300, the CP/M emulator, and their new $219 graphic/ EPSON-like printers. They also gave out their new **ST Software Catalog**, 347 pages of ST software for $10.

As mentioned last month, there were many accounting and other business packages their by a number of vendors.

Abacus POB 7211, Grand Rapids, MI 49510) had a separate booth with many ST books. As you know, the ST was released in Germany quite a bit before it was here, and is now the fastest selling computer there. A team from this company started writing this multi-volumn encyclopedia for the ST and translated the books into English. They gave me four of them; there are many more announced. I'll report on these books in a separate article. They also had the developer of their **PC-Board Designer** demonstrating this professional level CAD package ($400) which was very nice. You first define the components for the board, define the connections, put the components on the screen where you want them and the program automatically draws every trace and connection of the layout on the screen. You then print it out on an Epson-like printer, reduce it to half size, and it is camera ready for photo-etching.

I also had the great pleasure to meet Sig Hartman, and found him to be a very warm and friendly person, eager to help user groups and willing to listen. Many of the vendors were previous ACE members, and it was fun meeting so many old and new friends.

The main reason for going was to receive the ANTIC! award from Jim Capparell. Jim is the publisher of ANTIC! and this was their 4th anniversity party, so they decided to give awards. The dinner was well attended and a lot of fun. We all had a good time, the awards were beautiful, and it was an honor to accept the award on behalf of all you ACE members!

With all this new software out, I imagine that there will be bugs. The problem is not the minor ones, but how quickly they are updated, how responsive the company is, and if they charge for bug fixes rather than updates with new features. I have also heard complaints from ST owners about some of the companies; hopefully these are just growing pains and not company policy. If any of you have used any of the above-mentioned hardware or software, or other new software, either 8 or 16 bit, we will all love to see your reviews. Send them on a disk, and we will return the disk with programs of your choice. [If you can, send them on a 3.5 disk in ASCII format (non-WP mode on 1st-Word, or ST Writer format. We edit the newsletter on an ST. — J.B.]

The disk library continues to expand. Please look at the last few months of back issues, and see if you might like to order any disks. **If everyone in the club orders only one disk, our cash flow problem will be over — see if you can find something you would like.** Jim Bumpas will report on the ST library; here are some new additions to for the 8-bit machines. They are from other clubs, and have some of the same programs in previous ACE disks, but may be updated or better documented.

Thanks to Paul Freeman of B.A.C.E. (Baltimore):

BACE #28 — Graphics Utilities. A collection of drawing, charactor set generators, PMgraphics, and other utlities, including a graphic generator for Bulletin Board Systems. $5.

BACE #39 — A nice Checkbook program. $5.

BACE #40 — Koala Pictures. $5

BASE #53 — VT-100 Terminal Emulator — a public domain product from Compuserve/ANTIC!. A very nice DEC and IBM 3270 terminal emulator which produces 80 column on the screen of any decent monitor (not a TV), just a terminal — no downloading, etc. capabilities. Works with most modems, including Atari, MPP, AID interface, and 850 interface. If you need to call a mainframe, this is it! Well documented, 2 sides, $10.

BACE #63 — The Graphics Trilogy. A freeware double sided disk with a character generator, a drawing program and a multimode maker. Very nicely done — it is doublesided, but since it is a freeware product, there is only a copying charge of $5; Tim would like $9 if you like the program.

BACE #65 — FigFORTH 1.1. The SPACE FORTH we have, expanded and documented. 2 sides, $10.

From S*P*A*C*E (Seattle):

SPACE #10 — Printer Utilities-a 2 sided disk with many printer utilities for most all printers; dumps, print Atari Graphic charactors, convert PrintShop or other pictures, etc. $10.

SPACE #24/25 — TELE-COMM — a double sided disk with many public domain communication programs for most modems. Similiar to the ACE disk. $10.

SPACE #27 — Disk Utilities — a double sided disk with a wide variety of disk tools, copy programs, and some special files for PERCOM owners. $10.

The new edition of David Lien's classic book, **The BASIC Handbook** (Compusoft,535 Broadway, El Cajon, CA 92021, $25) is now out, This massive (861 pages) 3rd Edition covers almost all BASIC's available and is very useful in trying to convert programs. Unfortunately, it no longer has an Atari (8-bit) section like the 2nd Edition had, and nothing on the ST BASIC, but does explain what keywords mean in a variety of other BASICS. Since ST BASIC is close to MicroSoft, should be similiar to the book, but Atari 8-bit is much different — get the 2nd edition if you have an 8-bit Atari.

BUMPAS REVIEWS

ST: F.A.S.T.E.R. newsletter (c/o Andre Lafreniere, 1161 Des Fauvettes, Boucherville, Quebec, Canada J4B 6A8) appears in May, 1986 with vol. 1 no. 1. This is a very cleverly designed disk-based newsletter which is completely integrated into the GEM desktop. All the articles, pictures (pictures!), and programs are available from the desktop menus when the newsletter program is run. The first issue contains a review of the Michtron BBS, a list of software available from FASTER, text and program files for TINYTOOL.ACC and BICALC.ACC, among other goodies. I enjoy seeing the manner in which they have put together this effort. I believe you will, too.

TIME BANDIT

ST: TIME BANDIT ($40, MichTron) describes itself as the "ultimate" arcade game. It may well be. It's really 16 games in one, not all of which are strictly arcade games. They might more accurately be called "arcade adventure" games. You'll find things reminiscent of many of the great arcade games of the past: Pac Man, Centipede, Six-Gun Shoot-out, Star Trek, and others I don't know. 16 of them, each different. This game also has another amazing feature: 2 players may play simultaneously, and may interact on the screen. Play action may be by keyboard, mouse, or joystick. I find my 8-bit background makes the joystick easiest for me. Minimal use of the keyboard for verbal responses, is required.

Each of the 16 "time gates" send you to a different world with 16 levels of play. 256 game levels, each with several screens. You may never need to buy another arcade game for the rest of your life. This one could keep you occupied.

As you progress up the scale of levels, the action not only becomes faster and more ferocious, but in many of the modules additional clues will appear and opportunities to use them to solve various puzzles which present themselves. I got involved quite a bit with the Star Trek module. Finding various objects and carrying them to other parts of the ship to get it to working order. Many of the modules have similar action: Coming up to something which prevents you from proceeding; finding an object; moving it and using it to cross the barrier.

I can recommend this game to anyone who enjoys arcade adventure games. The game is copy-protected, but the documentation contains a very unusual description of this protection. It's an apology! And I believe it's sincere. Who ever heard of a software company being apologetic for having to protect its software against piracy? Their warranty is one of the best in the industry, too: If the disk becomes defective within 30 days, replacement is free. Any time thereafter, a $5 charge for replacement will get you a new disk.

MUSIC STUDIO

ST: **Music Studio** (designed by Audio Light and marketed by Activision, $60?) gives one the most power over the music capabilities of the ST I have yet seen. The program will also be available for the XE. One may use keyboard, joystick, mouse or touchpad with the program. Numerous demonstration songs are available for you to play and "fiddle" with.

This program gives you much of the power a word processor gives you over text documents. Block operations are simply executed to move, copy or delete strings of notes. Blocks may also be used to try out different instruments on those chosen notes (you can make instrument changes for the entire song, too).

One may design instruments which are not among the 16 given in the program. To execute this feature one is told "all the dynamics of a sound synthesizer are right at your fingertips." There are three different screens full of menus (some of which open new screens) or selections you can make on screen for easy, intuitive access to all the functions of the program.

I am not a musical person at all, but I was able to take a piece of sheet music and copy it into the computer and PLAY the piece. It even sounded right! The first time! I didn't come across any musical notation on the sheet music which I could not reproduce on screen in this program. A sweep of the mouse will place or move (or remove) any note or notation. Painless music, no years of finger-breaking practice required. Of course, those who are dedicated to music will be able to do much more with it than me.

The program will control MIDI compatible electronic instruments, or you may use the ST sound chip. You can add lyrics to your song file, and a graphics printer (I use an SG10) will print out any music composition. This program will be great fun for anyone. Anyone beginning a study of music can find this program a magic door into most of the intricacies of musical notation and experimentation. Even serious musicians may prefer this program for a quick, easy and inexpensive way to quickly test out many variations on a theme they may be developing.

GENIE

The Atari ST seems to be contributing a lot to the development of 8-bit software for the XL/XE line also. **GENIE** ($30, New Horizons Software, Box 180253, Austin, TX 78718) seems to be such a development.

Do you envy those ST owners with their desktop accessories? Well, envy no more. This program gives XL/XE owners some "desktop" accessories of their own. Actually, these accessories reside "under" the OS in the unused 16k of the 64k machines.

While you are inside any other program, a mere 3 simultaneous keystrokes away is a Note Pad which can hold more than 100 lines of text; a 5-Function Calculator with memory and a choice of calculations in hexadecimal or decimal; an Atascii table; and some limited DOS functions for file management.

The manual is superfluous, but well-done and complete. The program performs effortlessly and with an intuitive feel which will make you feel right at home. There are some limitations: The program will not work with any other program which also makes use of that 16k area. Also any program which modifies the keyboard handler in such a way that the keys used by Genie are changed, will not work. I recommend this program for anyone who isn't ready to buy an ST yet in order to have some desktop accessories, or for ST owners who also want some modern features on their older 8-bit machines.

— Jim Bumpas

I'M IN LOVE

With Atari ST

(sing to the accompaniment of "I'm in Love with a Wonderful Guy" by R. Rogers, but only without Mr. Rogers' knowledge.)

I'm as witty as Buchwald and Bombeck.
I'm as proper as Emily V.
No more a sage staring down a blank page
since I purchased Atari ST.
I am in a perpetual frenzy
with a perpetual creativity.
Confidence drums through my fingers and thumbs
when I write with Atari ST.
Though I'm trite and as crass as a clerk
selling brass in a storm door to door,
quite didactic and stale as a fruitcake,
I fail to stop turning out more.
Life is lovely when you are a writer
and corrections come so easily.
So, if this sounds strange and a little deranged
I'm in Love, I'm in Love, I'm in Love, I'm in Love,
I'm in Love with Atari ST!

— Sherry L. Beaver
Eugene, Oregon

HINTS

For New Users

DOUBLE DOS 2.0 (Reprint: LACC, April 1986)

You can make a double density DOS 2.0S disk with SmartDOS and two disk drives. First boot the SmartDOS and format a disk in drive 2 as double density. Now turn off the computer. Wait 12 seconds if you have the 256k mod. Or boot a DOS 2.0S disk immediately if you don't. Now, use the H option to write the DOS files to the disk in drive 2. In a matter of seconds you will have a double density DOS 2.0S. Your friends will love it and want the recipe.

STuff

JUNE INSOFT

The June, 1986 issue of InSoft (1834 Beacon St., #1, Brookline, MA 02146) magazine contains a good selection of news, executable programs and document files. They are also continuing their series on 3D projections. They are also organizing a purchasing club. For $50 you can join and buy ST merchandise for up to 40% off. Try a double-sided drive for $205. Their catalog is also on the disk and contains hundreds of items of hardware and software. If you join before July 15, 1986 you can get additional discounts.

TOUCH ME

Over $2,000 in prizes to be awarded by The Touching Poet, Antic magazine, Analog magazine, and Batteries Included. For more information about the Atari ST "Touch Me" Art Contest, write: The "Touch Me" Contest, The Touching Poet, 4339 North 39th Street, Phoenix, AZ 85018, (602) 955-1148.

ST Library

New additions this month include two items to the Telecommunications disk: XMOTERM.TOS is a smoothly-working terminal program which I find very easy to use. ADDRESS.PRG and .DOC will keep a list of addresses and perform two kinds out output. You can use the program to print address labels. You can also use the program as a phone dialer. Two great utilities which appear on the June, 1986 issue of InSoft Magazine (disk).

Hippopotamus Software has also provided our public domain library with a disk which demonstrates several of their programs for the ST. Ask for it if you want to see it.

We have some disks in our ST library which I find myself using all the time. The Word Processing disk contains version 1.4 of ST Writer which the Mind Mine people have improved with Atari's permission. I use this program to edit the ACE Newsletter every month.

The Desktop Stuff disk contains several accessories I keep on my desktop all the time, including calculator, ramdisk and calendar.

I use the VIP spreadsheet for more and more things everyday. Many of the templates on our VIP.WKS disks are handy for keeping records and manipulating data of all kinds.

If you don't already have these kinds of programs, you'll find your personal productivity much improved by their addition to your library. By the way, if you contribute a public domain program to our library, together with an article describing it, you may have your choice of any disk in our library in return.

ALL GAMES COURTESY OF PAGE 6

SOLAPAK

[A response to our SolaPak review] . . .The disk contains 13 text files with over 64000 characters and still does not contain enough detail to fully describe all the possibilities of SOLAPAK, nor is it as clear as it could be in instructing the novice user. For this reason the documentation is constantly being upgraded, mainly due to feedback from users such as Mr. Rogers.

. . .SOLAPAK comes with [installation] templates for 8 popular printers and as Mr. Rogers found, like many SOLAPAK purchasers, one of these templates is likely very close to that of the desired printer. What . . .has since been added is that you can skip the printer customization and do it at a later time . . .

Mr. Rogers finds it inconvenient to edit a configuration file in order to change the size of the ramdisk or some other option. As mentioned in the documentation for version 1.0, removal of the definition line for an item such as ramdisk size, screen saver enable, and other selectted features will cause you to be prompted for the item each time you reboot. This eliminates the need to load an editor just to change the size of a ramdisk. Documentation now includes a section on usage tips which describes these capabilities in more detail than the version Mr. Rogers received.

. . .The maximum number of files which can be queued waiting to be printed is 8. The maximum number of files which can be contained in the ramdisk is 128 . . .with 32 a suggested default . . . Most print spoolers reserve a RAM buffer for holding a single file to be printed and then won't let you use that RAM if no printing is being done. SOLAPAK conrtains no such buffer and will print any file which is in its ramdisk. When printing is not being performed you have full use of the RAM in the form of the ramdisk.

Contrary to Mr. Rogers' statement, SOLAPAK can be used with 1st-Word. The trouble arising with both 1st-Word and ST Writer is that these programs store disk files in their own unique formats. These formats are converted automatically to printable form by the programs as they output to the printer. Trying to print these unique formats with SOLAPAK produces undesirable results. What is both necessary and a nuisance is that these unique formats must first be converted to standard ASCII files prior to printing with SOLAPAK. However, the pain is eased by using the ramdisk as a temporary location for these ASCII files because of the speed with which the files can be produced.

We're glad Mr. Rogers finds the price reasonable. In keeping with the motto of "Power without the Price" we have offered for sale 3 functions at an average price of $10 each. In addition our update policy of $3 supports our commitment to low cost quality software.

Sincerely,

Tim G. Hunkler
1807 N. Evergreen
Chandler, AZ 85224
(602) 899-6992
CompuServe #76064,1172

KEYBOARD HELP

Reprint: WAND, April 1986

Having problems with your XL/XE keyboard? In particular, is it broken (will not type certain letters or numbers)? Well, help is here with a capital H. If you have one of these new beasts with its membrane (microwave oven) disguised keyboard and it breaks, how in ±];* do you fix it??!! Actually the answer is simple. First DO NOT solder (attempt, that is) it. It will not solder and you are wasting your time and may even make it unrepairable. Go to your local electronics shop. Ask the store if they carry "GC" products. In particular, conductive paint. The kind I have is old and is actually silver paint (not color — has silver particles in it). The new conductive paint must have a different conductor because I paid $10 for a very small bottle and the new stuff is less than half that price.

Now the hard part. Find a good friend with a multimeter and find the break in the clear membrane. Once you have the break located, paint a small amount of the conductive paint over the break. Make sure not to paint several traces together. If you do, scrape off the paint between the traces and check with the multimeter for shorts. Let the keyboard dry for awhile, test it with the meter to make sure it's fixed, then put it all back together. Easy, right? Right!

— Gerry Stotts

[WAND editor notes: This repair method is also useful for repairing traces on printed circuit boards.]

LANDSCAPE

LANDSCAPE
BY MALCOLM IREDALE

Those alien invaders are at it again! This time descending one by one to the Landscape below. You can blast off and destroy them but you have only limited missiles and fuel.

Use a joystick in port 1 to control your ship so that it is below an alien and press the trigger to fire. Avoid contact with the aliens and the landscape at all times as this is lethal. You eoll also lose lives if you allow five aliens to land or if your fuel runs out, but you can re-fuel by shooting the diamond shaped fuel store which appears occasionally. This will also replenish your missles.

Once all the aliens have been destroyed you must return quickly to base to prepare for the next level.

```
0 REM ****************
1 REM ** 1-8-83.  **
2 REM ** LANDSCAPE **
3 REM **     BY    **
4 REM ** M.IREDALE **
5 REM ****************
6 I=0
7 M5=12:FL=500:DIM M$(64),A$(8),B$(4):
LV=3:DIF=3
9 OPEN #1,4,0,"K:"
10 GRAPHICS 1+16:POKE 752,1:SETCOLOR 2
,0,0:POKE 708,198:POKE 709,218
20 DL=PEEK(560)+256*PEEK(561)
30 POKE DL+3,66
31 POKE DL+6,2
32 POKE 87,1:POSITION 4,4:? #6;"initia
lizing":GOSUB 32400
33 GOSUB 23000:POKE 87,0:POSITION 1,0:
? "FUEL:";FL:POSITION 31,0:? "MSLs:";M
5;
34 POSITION 13,0:? "POINTS:00000";SC:P
OSITION 18,1:? "HI:";HI
35 POSITION 1,1:? "Lives: ( ( (";
36 POSITION 30,1:? #6;"LANDED:0":POKE
87,1
37 FOR L=0 TO 19:POSITION L,4:? #6;CHR
$(ASC("*")-32);:NEXT L
39 POSITION 1,8:? #6;"◢ ▙▙▙▙▙your shi
p"
40 POSITION 1,10:? #6;"▗ ▙▙▙your missi
le"
41 POSITION 1,12:? #6;"▗ ▙▙▙▙▙your ba
se"
42 POSITION 1,14:? #6;"▝ ▙▙▙▙▙▙▙▙▙ali
en"
43 POSITION 1,16:? #6;"▝ ▙▙▙▙▙▙▙▙▙bon
us"
44 POSITION 1,18:? #6;"$ ▙▙▙▙▙landsca
pe"
48 GG=8:M$="hold down fire button to
start"                ":GOSUB 25000:IF
STRIG(0)=1 THEN 48
60 GOSUB 23000:GOSUB 30000
63 REM ************************
64 REM **** INITIAL REMARK ****
65 REM ************************
66 GG=5:M$="landscape" by malcolm ire
dale"                 ":GOSUB 25000
67 GG=0:M$="level ┌ to ┼ ":GOSUB 25000
:GET #1,A
68 IF A=49 THEN DIF=1
69 IF A=50 THEN DIF=2
70 IF A=51 THEN DIF=3
71 IF A<49 OR A>51 THEN 67
72 GOSUB 23000:GOSUB 30000
83 GG=0:M$="                    ":GOSUB 2
5000
84 POSITION 3,5:? #6;"────────────";

85 GOSUB 19000
86 XX=X:YY=Y
88 REM ****************************
89 REM *** MAIN MOVEMENT ROUTINE ***
90 REM ****************************
100 S=STICK(0)
101 G=G+1:IF G=80 THEN G=0
102 IF G=0 THEN POSITION 1,6:? #6;",":
POSITION 18,6:? #6;","
103 IF G=20 THEN POSITION 1,6:? #6;" "
:POSITION 18,6:? #6;" "
104 REM CANCEL MSLs IF STRIG(0)=0 AND
FLAG1=1 AND M5>0 THEN FLAG1=0:POSITION
 MX,MY:? #6;" ":SOUND 0,0,0,0:GOTO 106

105 IF STRIG(0)=0 AND FLAG1=0 AND M5>0
 THEN FLAG1=1:M5=M5-1:MX=X:MY=Y
106 IF FLAG1=1 THEN GOSUB 27011
107 IF FLAG1=1 THEN GOSUB 27011
109 IF FLAG2=0 THEN FLAG2=1:AY=5:AX=IN
T(RND(0)*14)+3:LOCATE AX,AY,S5:IF S5=3
2 THEN FLAG2=0:GOTO 112
110 IF FLAG2=1 THEN GOSUB 27500
112 X=X+(S=7)-(S=11)+(S=6)+(S=5)-(S=10
)-(S=9):Y=Y+(S=13)-(S=14)-(S=6)-(S=10)
+(S=9)+(S=5)
113 IF X<0 THEN X=19
114 IF X>19 THEN X=0
115 IF Y<7 THEN Y=7
116 IF Y>22 THEN Y=22
118 FL=FL-((DIF*2)-1):IF FL<=0 THEN FL
=0:C=1:GG=3:M$="you ran out of fuel
                ":GOSUB 25000:GOSUB 24000
130 LOCATE X,Y,SPOS:LOCATE X,Y+1,D:IF
SPOS<>32 THEN GOTO 20000
135 REM
136 IF D=32 THEN DD=1
137 IF D<>32 THEN SOUND 1,0,0,0
140 POSITION X,Y:? #6;"◢":IF S<>15 THE
N POSITION XX,YY:? #6;" ";
141 IF DD=1 THEN DD=0:POSITION X,Y+1:?
 #6;"_":SOUND 1,25,0,6:POSITION X,Y+1:
? #6;" "
145 XX=X:YY=Y
150 GOTO 100
17999 REM ********************
18000 REM *** NO LIVES LEFT ***
18001 REM ********************
18004 FOR V=0 TO 3:SOUND V,0,0,0:NEXT
V
18005 I=1
18010 POKE 87,1:GG=5:M$="sorry but yo
u have lost all your lives"
    ":GOSUB 25000
18011 IF SC>HI THEN GG=5:M$="you have
 attained a new high score"
    ":GOSUB 25000
18012 IF SC<HI THEN GG=5:M$="you did
not manage to reach a new high score"
                ":GOSUB 25000
18015 IF SC>HI THEN HI=SC
18016 GG=5:M$="press fire button for
another go"              ":V=1:GOSUB
25000
18017 IF STRIG(0)=1 THEN GOTO 18016
18020 LV=3:FLAG1=0:FLAG2=0:M5=12:FL=50
0:SC=0:LND=0:POP :GOTO 33
18999 REM ************************
19000 REM *** INITIAL SAUCER POS. ***
19001 REM ************************
19010 X=INT(RND(0)*17)+1
19020 FOR K=23 TO 10 STEP -1
19030 LOCATE X,K,LL:IF LL<>32 THEN POS
ITION X,K-2:? #6;"◢":Y=K-2:POSITION X,
K-1:? #6;CHR$(ASC("/")-32):RETURN
19040 NEXT K:RETURN
20000 REM ************************
20001 REM ***CHECK FOR SAUCER HIT***
20002 REM ************************
20010 IF SPOS=8 THEN GOTO 135
20011 IF SPOS=36 OR SPOS=37 OR SPOS=38
 OR SPOS=39 OR SPOS=164 THEN X=XX:Y=YY
:C=1:GOSUB 24000:GOTO 135
20015 IF SPOS=15 THEN GOSUB 21000:GOTO
 135
20016 C=1:HT=HT+1:GOSUB 24000:GOTO 135

20020 GOTO 20020
20999 REM ********************
21000 REM *** SAUCER ON BASE **
21001 REM ********************
21002 IF HT>=14 THEN GOSUB 25050
```

LAND CON'T

```
21010 X=XX:Y=YY:RETURN
23000 REM ************************
23001 REM **CLEAR BOTTOM OF SCREEN**
23002 REM ************************
23010 POKE 87,1:FOR W=6 TO 22:POSITION
 0,W:? #6;"                    ";:NEXT
 W:RETURN
23999 REM ********************
24000 REM *** LIVES UPDATE ***
24001 REM ********************
24002 FL=500:MS=12:POKE 87,1:POSITION
X,Y:? #6;"":FOR Q=0 TO 20:SOUND 0,25,
0,15:NEXT Q:SOUND 0,0,0,0:LV=LV-1
24003 POKE 77,0:SOUND 2,0,0,0:SOUND 1,
0,0,0
24004 MSL=30:FL=500
24005 IF LV=2 THEN FLAG1=0:FLAG2=0:GG=
5:M$="you have just lost a life"
          ":GOSUB 25000
24006 IF LV=1 THEN FLAG1=0:FLAG2=0:GG=
5:M$="you have just lost another life
"              ":GOSUB 25000
24007 IF LV=1 THEN M$="you are now on
 your final life"              ":GOS
UB 25000
24008 IF LV=0 THEN 24013
24009 IF HT>=14 THEN GOSUB 24050
24010 IF C=1 THEN C=0:GOSUB 23000:GOSU
B 30000:GOSUB 19000:POSITION 3,5:? #6;
"──────────":LND=0:GOSUB 24050:GOT
O 24012
24011 GOSUB 23000:GOSUB 30000:POSITION
 3,5:? #6;"──────────":LND=0:GOSUB
 24050
24012 IF LV=2 THEN POKE 87,0:POSITION
1,1:? "Lives: ( (  ";:POKE 87,1:RETURN

24013 IF LV=1 THEN POKE 87,0:POSITION
1,1:? "Lives: (    ";:POKE 87,1:RETURN

24014 IF LV<=0 THEN POKE 87,0:POSITION
 1,1:? "Lives: ZERO ";:POP :GOTO 18000

24040 REM ********************
24050 REM *** SCORE UPDATE ***
24051 REM ********************
24060 A$="00000":A$(6-LEN(STR$(SC)))=S
TR$(SC)
24065 B$="000":B$(4-LEN(STR$(FL)))=STR
$(FL)
24070 POKE 87,0
24075 POSITION 21,0:? A$:POSITION 36,0
:? MS;" ":POSITION 6,0:? B$
24076 POSITION 30,1:? #6;"LANDED:";LND
:IF LND>=5 THEN LND=0:C=1:GOSUB 24000:
RETURN
24080 POKE 87,1:RETURN
24998 RETURN
24999 GOTO 24999
25000 REM ************************
25001 REM *** SCOLL BOTTOM LINE ***
25003 REM ************************
25004 FOR L=1 TO LEN(M$)
25005 IF 17-L<2 THEN F=F+1
25010 POSITION 17-L+F,23:? #6;M$(1+F,L
)
25015 FOR G=0 TO GG:NEXT G
25020 NEXT L:F=0:RETURN
25047 REM ************************
25048 REM **SAUCER ON BASE MESSAGE**
25049 REM ************************
25050 FL=500:MS=12:FOR I=0 TO 3:SOUND
I,0,0,0:NEXT I:GOSUB 24050
25055 IF HT>=14 THEN GG=3:M$="well do
ne! you are going up a level"
       ":GOSUB 25000
25056 IF HT>=14 THEN IF DIF<3 THEN DIF
=DIF+1
25059 IF HT>=14 AND DIF=2 THEN GG=3:M$
="you are now on skill level two"
            ":GOSUB 25000
25060 IF HT>=14 AND DIF=3 THEN GG=3:M$
="you are now on skill level three"
            ":GOSUB 25000
25061 IF HT>=14 THEN HT=0:GOSUB 23000:
MSL=30:FL=500:LND=0:POP :GOSUB 30000:G
OTO 70
27000 REM **SAUCER MISSILE MOVEMENT**
27011 MY=MY-1:IF MY<=5 THEN FLAG1=0:PO
SITION MX,MY+1:? #6;" ":SOUND 2,0,0,0:
GOSUB 24050:RETURN
27012 SOUND 2,25-MY,8,15
27014 LOCATE MX,MY,MPOS:IF MPOS=32 THE
N 27020
27015 GOSUB 27200:IF MPOS=ASC(",") THE
N MS=12:FL=500:SC=SC+500:G=19
27016 IF MPOS=ASC("-")-32 THEN SC=SC+1
00:HT=HT+1:FLAG2=0:FLAG1=0:SOUND 2,0,0
,0:GOSUB 24050
27020 POSITION MX,MY:? #6;CHR$(9):POSI
TION MX,MY+1:? #6;" "
27025 IF FLAG1=0 THEN POSITION MX,MY:?
 #6;" "
27030 RETURN
27200 POSITION MX,MY:? #6;CHR$(ASC("+"
)-32)
27220 FOR H=1 TO 20:SOUND 2,46-H,0,15:
NEXT H:POSITION MX,MY:? #6;" ":RETURN

27300 POSITION X,Y:? #6;CHR$(ASC("+")-
32)
27320 FOR H=1 TO 20:SOUND 2,46-H,0,15:
NEXT H:RETURN
27499 REM ********************
27500 REM ***ALIEN MOVEMENT ***
27501 REM ********************
27505 AAY=AY:AY=AY+1
27510 LOCATE AX,AY,APOS
27520 IF APOS=36 OR APOS=37 OR APOS=38
 OR APOS=39 OR APOS=15 THEN POSITION A
X,AY-1:? #6;" ":FLAG2=0:F=1:HT=HT+1
27522 IF F=1 THEN F=0:LND=LND+1:SOUND
0,50,8,15:POSITION AX,AY-1:? #6;"+":GO
SUB 24050:SOUND 0,0,0,0:POSITION AX,AY
-1:? #6;" ":RETURN
27525 IF APOS=ASC(")")-32 THEN FLAG1=0
:FLAG2=0:POSITION AX,AY:? #6;" ":MY=MY
-1:GOSUB 27200:SOUND 2,0,0,0:HT=HT+1
27526 IF APOS=ASC(")")-32 THEN GOSUB 2
4050:RETURN
27530 POSITION AX,AY:? #6;"":POSITION
 AX,AAY:? #6;" ":RETURN
29999 REM ****************
30000 REM *** LANDSCAPE ***
30001 REM ****************
30002 HT=0
30055 Y=INT(RND(0)*6)+13:K=-1:YY=20
30056 FOR D=0 TO 19
30057 IF YY=Y OR YY>=22 OR YY<=7+((4-D
IF)*3) THEN K=-K:Y=INT(RND(0)*6)+13:D=
D-1
30058 YY=YY+K
30060 IF K=-1 THEN POSITION D,YY:? #6;
CHR$(INT(RND(0)*2)+36):POSITION D,YY+1
:? #6;"":POSITION D,YY-1:? #6;" "
30070 IF K=1 THEN POSITION D,YY:? #6;C
HR$(INT(RND(0)*2)+38):POSITION D,YY+1:
? #6;"":POSITION D,YY-1:? #6;" "
30080 NEXT D:RETURN
32400 REM ************************
32401 REM ***CHARACTER DEFINITION***
32402 REM ************************
32420 POKE 106,PEEK(106)-8
32430 P=PEEK(106)
32432 N=P*256
32434 L=1023:FOR M=0 TO 1023:L=L-1:POS
ITION 9,5:? #6;INT(L/30);" ":POKE N+M,
PEEK(57344+M):NEXT M
32435 POSITION 9,5:? #6;"  ":POSITION
4,4:? #6;" energizing ":FOR J=4 TO 15
32436 FOR M=0 TO 7:READ D:POKE N+M+(J*
8),D:NEXT M:NEXT J
32635 POKE 756,P
32636 RETURN
32637 REM ************************
32638 REM **DATA FOR NEW CHARACTERS**
32639 REM ************************
32640 DATA 9,94,236,72,80,96,64,128
32642 DATA 1,2,4,8,16,32,64,128
32644 DATA 144,122,55,18,10,6,2,1
32646 DATA 128,64,32,16,8,4,2,1
32648 DATA 0,24,126,255,126,36,36,0
32650 DATA 0,0,24,24,24,24,24,0
32652 DATA 0,0,0,0,204,204,51,51
32654 DATA 42,148,75,186,93,146,41,84
32656 DATA 16,40,68,146,68,40,16,0
32658 DATA 0,130,146,170,214,170,146,1
30
32659 DATA 16,40,40,84,84,170,84,0
32660 DATA 255,255,129,66,36,24,92,231
```

NUTTY

NUTTY
by Andrew Starkie

Nutty is a good name for this game! Not only do you have to catch falling nuts but the game title- Squirrel on a Pogo Stick- has to be one of the nuttiest around!

The object is simply to move your squirrel right or left to catch falling nuts. You lose a life each time you allow a nut to fall to the floor.

Hopefully this little program will give you some ideas for writing your own simple games without getting into the complex programming areas. It's simple but it's still fun.

```
1 REM ********************************
2 REM *  SQUIRREL ON A POGO STICK.  *
3 REM *  BY ANDREW STARKIE . 1984.  *
4 REM *  J.D.                       *
5 REM ********************************
8 Q0=0:Q1=1:Q3=3:Q4=4:Q5=5:Q6=6:Q7=7:Q
9=9:Q10=10:Q11=11:Q13=8:Q18=18:Q31=31:
Q34=34:Q70=70:Q106=106:Q256=256
9 Q279=53279:Q500=500:Q559=559:Q756=75
6
10 GRAPHICS Q0:? "BLACKOUT COMING UP":
FOR A=Q0 TO Q500:NEXT A
11 POKE 106,PEEK(106)-5:ST=(PEEK(106)+
1)*256:POKE 559,0:GOSUB 49:GRAPHICS 0:
POKE 559,34:POKE 756,ST/256:XX=ST/256
12 ? "INITIALISING...%%%%%":? "
          $$$$⊞<":GOSUB 51:FOR A=Q0 T
O 1000:NEXT A
15 GRAPHICS Q18:POKE Q756,ST/Q256
16 ? #Q6;"SQUIRREL ON A POGO":? #Q6;"
     STICK":? #Q6:? #Q6;"          BY":
? #Q6;"  ANDREW STARKIE"
17 FOR P5=0 TO 18:POSITION P5,10:? #6;
"%":POSITION P5,11:? #6;"⊞":FOR A=0 TO
 70:NEXT A:POSITION P5,11:? #6;"⊞"
18 FOR A=0 TO 70:NEXT A:POSITION P5,Q1
1:? #6;" ":POSITION P5,Q10:? #Q6;" ":N
EXT P5
19 POSITION Q4,Q10:? #Q6;"PRESS start"

20 POKE Q279,Q0
21 IF PEEK(Q279)<>Q6 THEN 21
22 POSITION Q4,Q10:? #Q6;"  O.K.      "
:DIM POGO$(Q3):OPEN #Q1,Q4,Q0,"K:"
23 REM
24 POSITION Q1,Q10:? #Q6;"DIFFICULTY (
1-3)"
25 GET #Q1,Q:Q=Q-48:IF Q<Q1 OR Q>Q3 TH
EN 25
26 LEV=Q
27 FOR A=Q0 TO Q500:NEXT A:NUTS=Q0:LI=
Q3:POGO$=" # "
28 GRAPHICS Q18:POKE Q756,XX:POSITION
Q0,Q0:? #Q6;"NUTS=";NUTS:FOR A=LI TO Q
1 STEP -Q1:POSITION 20-A,Q0:? #Q6;"%";
:NEXT A
29 POSITION Q0,Q11:? #Q6;"............
........";:SQ=Q10:POKE Q559,Q34
30 POSITION SQ,Q9:? #Q6;"%":POSITION S
Q,Q10:? #Q6;"⊞"
31 NT=INT(RND(Q0)*Q18)+Q1:NUTDRP=Q1
32 ST=PEEK(632):POSITION Q0,Q0:? #Q6;"
nuts ";NUTS
34 IF ST<>Q7 AND ST<>Q11 THEN FOR A=Q0
 TO Q10:NEXT A:GOTO 37
35 IF ST=Q7 AND SQ<Q18 THEN SQ=SQ+Q1
36 IF ST=Q11 AND SQ>Q1 THEN SQ=SQ-Q1
37 POSITION NT,NUTDRP:? #Q6;" ":POSITI
ON NT,NUTDRP+Q1:? #Q6;"&":NUTDRP=NUTDR
P+0.2*LEV
38 POSITION SQ-Q1,Q9:? #Q6;" % ":POSIT
ION SQ-Q1,Q10:? #Q6;" ⊞ ":IF RND(Q0)>0
.5 THEN POSITION SQ-Q1,Q10:? #Q6;" ⊞ "

39 IF NUTDRP<Q7 THEN 32
40 IF NT=SQ THEN NUTS=NUTS+1:SOUND 0,5
0,10,10:FOR A=0 TO 10:NEXT A:SOUND 0,0
,0,0:POSITION SQ,8:? #6;" ":POSITION S
Q,7:? #6;" ":GOTO 31
41 POSITION NT,8:? #6;" ":POSITION NT,
10:? #6;"&":SOUND 0,255,13,15:FOR A=0
TO 99:NEXT A:SOUND 0,0,0,0:POSITION NT
,10:? #6;" ":LI=LI-1
42 POSITION NT,Q7:? #6;" ":POSITION 19
-LI,Q0:? #Q6;" "
44 IF LI=Q0 THEN POSITION Q5,Q5:? #Q6;
"game over":POSITION Q4,Q7:? #Q6;"PRES
S START"
45 IF LI>Q0 THEN GOTO Q31
46 POKE Q279,Q0
47 IF PEEK(Q279)<>Q6 THEN 47
48 GOTO 27
49 FOR A=Q0 TO 1023:POKE A+ST,PEEK(573
44+A):NEXT A:RETURN
51 FOR A=24 TO 63:READ B:POKE ST+A,B:N
EXT A:RETURN
53 DATA 124,124,124,124,56,16,16,16,12
4,124,56,56,16,16,16,16,192,192,216,21
2,252,248,248,60,0,60,110,70,94,126
54 DATA 60,0,255,255,0,255,255,0,255,2
55
```

C -

LEFT OUT LAST MONTH

```
BYTE FUNC PARSE(BYTE ARRAY S1,S2,S3)
;
; GIVEN A FILESPEC PARSES OUT EITHER
; THE DEVICE,NAME,TYPE,OR NAME.TYPE
;
; S1 -- RETURN STRING POINTER
; S2 -- FILESPEC POINTER
; S3 -- COMMAND POINTER
;
BYTE
 STATUS,COLON,PERIOD,LENGTH

 STATUS=0

 COLON=INDEX(S2,":")
 PERIOD=INDEX(S2,".")
 LENGTH=S2(0)

 IF 0#COLON AND 0#PERIOD AND 0#LENGTH
THEN
  IF EQS(S3,"DEVICE") THEN
   PRINTF("IN %S%E",S3)
   SCOPYS(S1,S2,1,COLON)
   PRINTE(S1)
  ELSEIF EQS(S3,"NAME") THEN
   PRINTF("IN %S%E",S3)
   SCOPYS(S1,S2,COLON+1,PERIOD-1)
   PRINTE(S1)
  ELSEIF EQS(S3,"TYPE") THEN
   PRINTF("IN %S%E",S3)
   SCOPYS(S1,S2,PERIOD+1,LENGTH)
   PRINTE(S1)
  ELSEIF EQS(S3,"BOTH") THEN
   PRINTF("IN %S%E",S3)
   SCOPYS(S1,S2,COLON+1,LENGTH)
   PRINTE(S1)
  ELSE
   PRINTE("ERROR IN PARSE, BAD COMMAND
.")
  FI
  STATUS=1
 ELSE
  PRINTE("ERROR IN PARSE, BAD FILESPEC
")
 FI

RETURN (STATUS)
```

MANDELBROT SET

CON'T FROM LAST MONTH

```
5 REM SCRENLOD.LST
6 REM REF COMPUTE OCT 85 P10
7 REM MY VERSION... THERE IS
8 REM A SAVE PROG TO GO WITH
9 REM THIS....
10 REM SCREEN LOAD ROUTINE SETUP
15 CLR
20 DIM A$(15),CREG(9):MODES=200:NAME=5
00:SCRENLOAD=17000
30 FOR A=1536 TO 1542
40 READ B:POKE A,B
50 NEXT A
60 DATA 104,104,104,170,76,86,228
90 REM
100 GOSUB MODES
110 GOSUB NAME
112 IF MODE()15 THEN GRAPHICS MODE
115 IF MODE=15 THEN GOSUB 300
120 GOSUB SCRENLOAD
130 GOTO 16000
200 ? :? "%WHEN YOU HAVE FINISHED VIEW
ING "
205 REM
210 ? "PRESS THE SELECT KEY TO LOAD "
220 ? "ANOTHER PICTURE. "
230 ? :? "WHICH GRAPHICS MODE "
240 ? :? "7 8 9 10 11 15";:INPUT MODE
260 IF MODE=7 THEN MODE=23
270 IF MODE=8 THEN MODE=24
290 RETURN
300 REM SET UP GR.7+ SCREEN
305 FOR R=0 TO 8:CREG(R)=PEEK(704+R):N
EXT R
310 GRAPHICS 8+16:DLIST=PEEK(560)+PEEK
(561)*256:POKE DLIST+3,78
315 FOR R=0 TO 8:POKE 704+R,CREG(R):NE
XT R
320 FOR CHANGE=DLIST+6 TO DLIST+204:IF
 PEEK(CHANGE)=15 THEN POKE CHANGE,14
330 IF PEEK(CHANGE)=79 THEN POKE CHANG
E,78
340 NEXT CHANGE:POKE 87,7:RETURN
490 REM
500 REM GET NAME ROUTINE
510 REM
520 ? :? "WHAT FILE DO YOU WISH TO SEE
?"
530 ? :? "TYPE D:FILENAME.PIC "
540 INPUT A$
560 RETURN
15990 REM
16000 REM WAIT FOR SELECT KEY
16010 REM
16020 POKE 53279,8
16030 IF PEEK(53279)=5 THEN GRAPHICS 0
:GOTO 10
16040 GOTO 16030
17000 REM
17010 REM SCREEN LOAD ROUTINE
17020 REM
17030 OPEN #1,4,0,A$
17040 POKE 852,PEEK(88):POKE 853,PEEK(
89):POKE 856,220:POKE 857,30:POKE 850,
7
17050 A=USR(1536,16)
17070 CLOSE #1
17080 RETURN
```

```
10 REM DATACHK
20 REM CHECKS THE ITERATION COUNT FOR
       MANDELZOOM (MANDLCAL)
30 REM by Dick Kellett
40 REM
50 REM MACE MARCH 1986
60 REM ***************
70 REM ***************
80 ? CHR$(125):DIM A(100),FILE$(15),DA
T$(8)
90 ? "ENTER NAME OF DATA FILE. DO NOT
USE   DEVICE OR EXT. PROGRAM PROVIDES
A D:  DEVICE AND A DAT EXT"
100 INPUT DAT$:TRAP 160
110 FILE$(1,2)="D:":FILE$(3,LEN(DAT$)+
2)=DAT$:FILE$(LEN(DAT$)+3)=".DAT"
120 CLOSE #1:OPEN #1,4,0,FILE$
130 FOR F=1 TO 100:A(F)=0:NEXT F
140 GET #1,N:A(N)=A(N)+1
150 GOTO 140
160 CLOSE #1
170 ? CHR$(125):FOR P=1 TO 100
180 HPOS=(P-1)*7:VPOS=0
190 IF HPOS>30 THEN HPOS=HPOS-35:VPOS=
VPOS+1:GOTO 190
200 POSITION HPOS,VPOS
210 ? P;"*";A(P):NEXT P
220 ? "PRESS OPTION FOR GRAPH. VERT AX
IS IS  SCALED TO A SQUARE LAW FOR COUN
TS"
230 IF PEEK(53279)=3 THEN 250
240 GOTO 230
250 GRAPHICS 8:V=0:SETCOLOR 2,0,0:COLO
R 1:POKE 657,0
260 ? "
"
270 ? "1----------50----------
-----100"
280 FOR V=1 TO 100:IF V=1 THEN PLOT V,
159-SQR(A(V))
290 DRAWTO V*3,159-SQR(A(V))
300 NEXT V
310 GOTO 310
```

```
10 GRAPHICS 0:POKE 710,208:POKE 712,24
:? :? ,"SCREEN DUMP":POKE 748,5
20 ? :? " GRAPHICS 7+ TO PX80 PRINTER"
30 ? :? "by Jerry White & Fernando Her
rera"
40 ? :? "Altered to PX80 and print eac
h ":? "register separately by Dick Kel
lett."
45 ? "WHEN PICTURE IS LOADED PRESS OPT
ION TO PRINT REGISTRATION LINE":? "PRE
SS START TO COMMENCE PRINTING"
50 ? :? "PLEASE WAIT. SCREEN DUMP LOAD
ING"
51 FOR TIME=1 TO 1000:NEXT TIME
60 GOSUB 1410
94 REM ***SCREEN DUMP****
95 REM *BY JERRY WHITE &*
96 REM *FERNANDO HERREA *
97 REM * ANTIC MAGAZINE *
98 REM * JANUARY, 1984  *
99 REM ******************
100 REM SDUMP 9/13/83
105 DIM PR1$(780),PR2$(780),HD$(4),FN$
(15),CIOV$(42),DL$(33),REG$(770)
110 TRAP 110:GRAPHICS 0:POKE 82,4:POKE
 752,1:POKE 710,160:POKE 712,24:? "ENT
ER DEV:FILENAME.EXT";:INPUT FN$
120 TRAP 40000
175 REM PRINT STRINGS
180 RESTORE 190:FOR ME=1 TO 4:READ IT:
HD$(ME,ME)=CHR$(IT):NEXT ME
190 DATA 27,76,2,3
200 PR1$=CHR$(0):PR1$(780)=CHR$(0):PR1
$(2)=PR1$:PR2$=PR1$
201 REG$(1)=CHR$(0):REG$(770)=CHR$(0):
REG$(2)=REG$(770):REG$(1,1)="7":REG$(7
70,770)="7"
210 PR1$(1,4)=HD$:PR2$(1,4)=HD$
220 REM DISK TO SCREEN SUBROUTINE
230 RESTORE 240:FOR ME=1 TO 42:READ IT
:CIOV$(ME,ME)=CHR$(IT):NEXT ME
240 DATA 104,104,104,10,10,10,10,170,1
04,104,157,66,3,104,157,69,3,104,157,6
8,3
250 DATA 104,157,73,3,104,157,72,3,32,
86,228,169,0,133,213,189,67,3,133,212,
96
260 REM DISPLAY LIST SUBROUTINE
270 RESTORE 280:FOR ME=1 TO 33:READ IT
:DL$(ME,ME)=CHR$(IT):NEXT ME
280 DATA 104,173,48,2,133,203,173,49,2
,133,204,160,1,200,177,203
290 DATA 201,15,240,4,201,79,208,4,233
,1,145,203,192,200,208,237,96
```

```
693 IF REG=2 THEN ? #1;"+ - - - -
 -  -  -  -  -  -  -  -  -  -  -+":F
OR LF=1 TO 4:? #1:NEXT LF
694 IF REG=3 THEN ? #1;"+ - - -
 - - - - - - - - - - - +":F
OR LF=1 TO 4:? #1:NEXT LF
695 IF REG=4 THEN ? #1;"+--------ALL R
EGISTERS--SHADES OF GREY----------+":F
OR LF=1 TO 4:? #1:NEXT LF
707 IF PEEK(53279)=3 THEN 707
708 IF PEEK(53279)=3 THEN 691
709 IF PEEK(53279)<>6 THEN 708
710 REM SETUP PRINTER FOR GRAPHICS
714 REM SETUP PRINTER FOR GRAPHICS
715 CLOSE #1:OPEN #1,8,0,"P:":? #1;CHR
$(27);"3";CHR$(18);CHR$(27);"L";CHR$(2
);CHR$(3);CHR$(27);CHR$(56)
720 PR1=ADR(PR1$):PR2=ADR(PR2$)
730 REM SCREEN TO PRINTER LOOP
740 FOR X=0 TO 39:PICT=191*40+SCR+X:ST
R=USR(1536,PICT,PR1,PR2):? #1;PR1$:? #
1;PR2$:NEXT X
750 ? #1:? #1:? #1;CHR$(27);CHR$(64):C
LOSE #1:POKE 764,255
790 GRAPHICS 0:POKE 710,208:POKE 712,2
4:? :? "PRESS OPTION TO ENTER NEXT REG
ISTER":? "PRESS SELECT FOR NEW PIC."
800 IF PEEK(53279)=3 THEN 301
810 IF PEEK(53279)=5 THEN 110
820 GOTO 800
930 CLOSE #1:GOTO 301
1400 REM SCREEN DUMP SUBROUTINE
1410 RESTORE 1420:FOR ME=1536 TO 1730:
READ ML:POKE ME,ML:NEXT ME
1420 DATA 104,104,133,204,104,133,203,
104,141,192,6,104
1430 DATA 141,191,6,104,141,194,6,104,
141,193,6,169
1440 DATA 1,133,207,169,191,133,208,16
0,0,177,203,141
1450 DATA 190,6,165,207,240,28,169,0,1
33,207,173,193
1460 DATA 6,24,105,4,141,193,6,133,205
,173,194,6
1470 DATA 105,0,141,194,6,133,206,76,9
5,6,169,1
1480 DATA 133,207,173,191,6,24,105,4,1
41,191,6,133
1490 DATA 205,173,192,6,105,0,141,192,
6,133,206,32
1500 DATA 137,6,32,146,6,32,160,6,32,1
37,6,32
1510 DATA 169,6,32,160,6,165,207,240,1
77,56,165,203
1520 DATA 233,40,176,2,198,204,133,203
300 REM COLOR REGISTER SELECTION
301 TRAP 301:? CHR$(125):POSITION 4,2:
? "WHEN PICTURE IS LOADED PRESS OPTION
 TO PRINT REGISTRATION MARKS"
302 ? "PRESS START TO PRINT"
303 POSITION 4,5
304 ? :? "ENTER REGISTER TO PRINT ":?
"0. DEF COL ORANGE":? "1. DEF COL GREE
N":? "2. DEF COL BLUE"
305 ? "3.BACKGROUND (REG 4).DEF COL BL
ACK"
306 ? "4. PRINT ALL REG IN SHADES OF G
REY":? "INPUT REG";:INPUT REG:IF REG<0
 OR REG>4 THEN 301
309 RESTORE REG*10+330
310 FOR ME=1731 TO 1746:READ PAT:POKE
ME,PAT:NEXT ME
329 REM COLOR REG.0  DEFAULT COL ORANG
E
330 DATA 0,0,0,0,15,15,15,15,0,0,0,0,0
,0,0,0
339 REM COLOR REG.1  DEFAULT COL LIGHT
 GREEN
340 DATA 0,0,0,0,0,0,0,0,15,15,15,15,0
,0,0,0
349 REM COLOR REG.2  DEFAULT COL BLUE
350 DATA 0,0,0,0,0,0,0,0,0,0,0,0,15,15
,15,15
359 REM BACKGROUND REG.4  DEFAULT COL
BLACK
360 DATA 15,15,15,15,0,0,0,0,0,0,0,0,0
,0,0,0
369 REM PRINT ALL REG. IN SHADES OF GR
EY
370 DATA 0,0,0,0,1,0,4,0,3,0,12,0,15,1
5,15,15
655 REM SETUP ANTIC 14 DISPLAY LIST
660 GRAPHICS 24:SCR=PEEK(88)+PEEK(89)*
256:ASM=USR(ADR(DL$))
670 CLOSE #1:OPEN #1,4,0,FN$:ASM=USR(A
DR(CIOV$),1,7,SCR,7680)
679 REM PRINT REGISTRATION POINTS
680 IF PEEK(53279)=3 THEN 690
681 IF PEEK(53279)=6 THEN 715
682 GOTO 680
683 TRAP 2030
690 CLOSE #1:OPEN #1,8,0,"P:":IF REG<4
 THEN ? #1;CHR$(27);CHR$(64);"
COLOR REGISTER ";REG:? #1
691 IF REG=0 THEN ? #1;"+- - - - -
 - - - - - - - - - - - +":F
OR LF=1 TO 4:? #1:NEXT LF
692 IF REG=1 THEN ? #1;"+ - - -
- - - - - - - - - - - +":F
OR LF=1 TO 4:? #1:NEXT LF
,198,208,165,208
1530 DATA 201,255,208,151,96,173,190,6
,41,3,10,10
1540 DATA 170,96,160,0,189,195,6,145,2
05,232,200,192
1550 DATA 4,208,245,96,173,190,6,74,74
,141,190,6
1560 DATA 96,160,0,189,195,6,10,10,10,
10,24,113
1570 DATA 205,145,205,232,200,192,4,20
8,238,96,0,0,0,0,0
1580 RETURN
2000 FOR S=1 TO 300:SOUND 0,5,10,15:SO
UND 0,0,0,0
2010 GRAPHICS 0:POKE 710,208:POKE 712,
24:POSITION 8,4:? "PRINTER NOT RESPOND
ING"
2030 POSITION 5,7:? "CHECK PRINTER AND
 PRESS START"
2040 IF PEEK(53279)<>6 THEN 2040
2050 TRAP 40000:GOTO 301
```

ACE'S ANNUAL PICNIC & SWAP MEET

JULY 9TH 7:00 P.M.

JASPER PARK

GRID

GRID

By Mark Hewson

Grid is a two player game requiring the use of two joysticks. After a lengthy initialisation (approx. 50 seconds) a grid appears and the players are offered a choice of four game options. The desired option is selected by pressing the trigger button as the option lights up.

The object of the game is to outscore your opponent by placing you coloured pieces strategically to either gain high scores yourself or to stop your opponent from obtaining a high score.

There are several "Mystery Squares" and if you place your piece on these there is an even chance of either doubling your score or scoring no points for that move. In the "Tricky" version of the game Mystery Squares have an additional function. If you choose a 'no score' square not only do you score zero for the move but your opponents piece will be placed on th square. "Tricky" games have red question marks over the Mystery squares and Normal games have green question marks.

```
0 ┘◆▌┘I
10 REM ******************************
11 REM *          GRID             *
12 REM * A TWO PLAYER STRATEGY GAME *
13 REM *          BY               *
14 REM *       MARK HEWSON         *
15 REM ******************************
16 REM
100 CLR :POKE 832,6
110 DIM TOTAL(2),P$(4),S(9),E$(10):GO=
1:P$="JKL"
120 S(1)=0:FOR I=2 TO 8:S(I)=I:NEXT I
130 GOSUB 2160
140 GOSUB 1080:POKE 54286,192
150 POKE 709,0:GOSUB 1940
160 TOTAL(1)=0:TOTAL(2)=0
170 GOSUB 1830
180 REM CURSOR
190 PA=96:PD=87:POKE 1536,0:POKE 704,1
4
200 RESTORE 220:FOR I=1571 TO 1578:REA
D X:POKE I,X:NEXT I
210 POKE 1611,7:POKE 53256,0
220 DATA 0,24,24,126,126,24,24,0
230 IF MOVES=0 THEN GOTO 610
240 POKE 53248,PA:POKE 1536,PD:FOR DE=
1 TO 50:NEXT DE
250 A=STICK(GO-1)
260 IF STRIG(GO-1)=0 THEN POKE 77,0:GO
TO 350
270 Z=PEEK(53279)
280 IF Z=6 THEN POKE 559,0:FOR X=53248
 TO 53251:POKE X,0:NEXT X:MOVES=0:? CH
R$(125):GOTO 150
290 IF A=15 THEN GOTO 250
300 IF A=13 AND PD<80 THEN PD=PD+8:GOT
O 240
310 IF A=14 AND PD>40 THEN PD=PD-8:GOT
O 240
320 IF A=11 AND PA>96 THEN PA=PA-8:GOT
O 240
330 IF A=7 AND PA<152 THEN PA=PA+8:GOT
O 240
340 GOTO 250
350 REM TRIG PRESSED
360 BONUS=1:A=(PA-48)/8:D=(PD-23)/8:LO
CATE A,D,Z
370 IF Z=91 THEN POSITION A,D:? #6;P$(
GO,GO):SOUND 0,10,10,14:SOUND 1,15,10,
14:GOTO 420
380 M=INT(RND(0)*2)
390 IF M=1 AND (Z=220 OR Z=203) THEN 1
660
400 IF M=0 AND (Z=220 OR Z=203) THEN 1
490
410 POKE 704,0:FOR Q=220 TO 255 STEP 0
.5:SOUND 0,Q,10,10:NEXT Q:SOUND 0,0,0,
0:POKE 704,14:GOTO 250
420 REM SCORE
430 N=0:S=0:A1=A:D1=D
440 LOCATE A1,D1,Z:IF Z=ASC(P$(GO,GO))
 THEN D1=D1-1:GOTO 440
450 LOCATE A1,D1+1,Z:IF Z=ASC(P$(GO,GO
)) THEN D1=D1+1:N=N+1:GOTO 450
460 S=S+S(N)*BONUS
470 N=0:A1=A:D1=D
480 LOCATE A1,D1,Z:IF Z=ASC(P$(GO,GO))
 THEN D1=D1-1:A1=A1-1:GOTO 480
490 LOCATE A1+1,D1+1,Z:IF Z=ASC(P$(GO,
GO)) THEN D1=D1+1:A1=A1+1:N=N+1:GOTO 4
90
500 S=S+S(N)*BONUS
510 SOUND 0,0,0,0:SOUND 1,0,0,0
520 N=0:A1=A:D1=D
530 LOCATE A1,D1,Z:IF Z=ASC(P$(GO,GO))
 THEN A1=A1-1:GOTO 530
540 LOCATE A1+1,D1,Z:IF Z=ASC(P$(GO,GO
)) THEN A1=A1+1:N=N+1:GOTO 540
550 S=S+S(N)*BONUS
560 N=0:A1=A:D1=D
570 LOCATE A1,D1,Z:IF Z=ASC(P$(GO,GO))
 THEN A1=A1-1:D1=D1+1:GOTO 570
580 LOCATE A1+1,D1-1,Z:IF Z=ASC(P$(GO,
GO)) THEN A1=A1+1:D1=D1-1:N=N+1:GOTO 5
80
590 S=S+S(N)*BONUS
600 TOTAL(GO)=TOTAL(GO)+S
610 GO=3-GO:AA=170:IF GO=1 THEN AA=78
620 POKE 656,0
630 POKE 657,6:? "VMTUO";
640 POKE 657,29:? "VMTUO";
650 POKE 656,1
660 POKE 657,8-INT(LEN(STR$(TOTAL(1)))
/2):? TOTAL(1);
670 POKE 657,31-INT(LEN(STR$(TOTAL(2))
)/2):? TOTAL(2);
680 IF MOVES=56 THEN POKE 53249,0:GOTO
 1190
690 POKE 53249,AA:POKE 53248,PA
700 IF MOVES=0 THEN POKE 1536,PD:FOR X
=10 TO 0 STEP -1:FOR I=100 TO 50 STEP
-4:SOUND 0,I,10,X:NEXT I:NEXT X
710 MOVES=MOVES+1:GOTO 250
720 REM TUNES
730 Q=12:D=10
740 IF RE=0 THEN RESTORE 840
750 IF RE=1 THEN RESTORE 860
760 D=D+16:READ S0,S1,S2,DE:POKE 540,D
E*Q
770 IF S0=999 THEN SOUND 0,0,0,0:SOUND
 1,0,0,0:SOUND 2,0,0,0:POKE 709,202:GO
TO 2070
780 SOUND 0,S0,14,10:SOUND 1,S1,10,10:
SOUND 2,S2,10,10
790 IF PEEK(540)=0 THEN 760
800 Z=PEEK(53279)
810 IF Z=6 THEN SOUND 0,0,0,0:SOUND 1,
0,0,0:SOUND 2,0,0,0:POKE 709,202:GOTO
280
820 D=D+16:IF D>255 THEN D=10
830 POKE 709,D:GOTO 790
840 DATA 0,40,50,2,40,50,60,2,50,0,40,
2,100,200,0,4,0,0,0,1,50,60,70,2,0,50,
100,2,130,230,50,2,200,120,30,4
850 DATA 999,999,999,0
860 DATA 150,120,0,2,100,80,0,2,90,75,
0,2,165,110,0,2,150,90,0,2,200,150,0,1
,75,0,0,0,200,150,120,4
870 DATA 999,999,999,0
880 REM CHOICES
890 IF STRIG(0)=0 OR STRIG(1)=0 THEN 8
90
900 POKE 656,0:POKE 657,1:POKE 755,2:?
 "Six    ?  twelve ?  six    !  twelve
 !";
910 POKE 656,1:POKE 657,1:? "normal @
 normal @  tricky @  tricky @";
920 POKE 1538,103:POKE 709,202
930 POKE 53250,52:POKE 706,144:SOUND 0
,125,10,4:MN=6:RE=0:GOSUB 970
940 POKE 53250,92:SOUND 0,100,10,4:MN=
12:RE=0:GOSUB 970
950 POKE 53250,132:POKE 706,80:SOUND 0
,75,10,4:MN=6:RE=1:GOSUB 970
960 POKE 53250,172:SOUND 0,50,10,4:MN=
```

GRID CON'T

```
12:RE=1:GOSUB 970:GOTO 930
970 FOR I=1 TO 3
980 POKE 656,3:POKE 657,12:? "
          ";
990 FOR DE=1 TO 15:IF STRIG(0)=0 OR ST
RIG(1)=0 THEN 1060
1000 NEXT DE
1010 POKE 656,3:POKE 657,12:? "trigger
 selects";
1020 FOR DE=1 TO 15:IF STRIG(0)=0 OR S
TRIG(1)=0 THEN 1060
1030 NEXT DE
1040 NEXT I
1050 RETURN
1060 ? CHR$(125):POKE 53250,0:POKE 706
,196:POKE 1538,131:POKE 709,202:POP :P
OP :POP
1070 SOUND 0,0,0,0:POKE 77,0:FOR DE=1
TO 80:NEXT DE:GOTO 2060
1080 REM BARS
1090 POKE 623,1:POKE 1613,7:POKE 1614,
7:POKE 53258,3:POKE 53259,3:POKE 53250
,0:POKE 53251,0
1100 POKE 706,0:POKE 707,52:POKE 1538,
103:POKE 1539,103
1110 FOR I=1591 TO 1598:POKE I,255:NEX
T I
1120 FOR I=1601 TO 1608:POKE I,255:NEX
T I
1130 REM ARROW
1140 POKE 705,119:RESTORE 1170
1150 FOR I=1581 TO 1588:READ X:POKE I,
X:NEXT I
1160 POKE 1612,7:POKE 1537,92
1170 DATA 60,60,60,60,255,126,60,24
1180 AA=78:RETURN
1190 REM FINISH
1200 Z=10:E$="    NUIX    ":C=136
1210 IF TOTAL(2)>TOTAL(1) THEN E$="UO
N XOSV ":C=72:GO=1
1220 IF TOTAL(1)>TOTAL(2) THEN E$="PUO
OS XOSV":C=202:GO=2
1230 POKE 710,C:POKE 755,1
1240 POSITION 5,9:? #6;E$
1250 FOR I=129 TO 13 STEP -2:SOUND 0,I
,10,12:SOUND 1,I-4,10,10:SOUND 2,I-7,1
0,10:SOUND 3,I-12,10,10:NEXT I
1260 FOR I=0 TO 3:SOUND I,0,0,0:NEXT I

1270 GOSUB 1420:GOSUB 1420
1280 D=5:FOR I=1 TO 3
1290 SOUND 0,100,10,Z:GOSUB 1420
1300 SOUND 0,75,10,Z:GOSUB 1420
1310 SOUND 0,50,10,Z:GOSUB 1420
1320 SOUND 0,75,10,Z:GOSUB 1420
1330 NEXT I
1340 SOUND 0,100,10,Z:GOSUB 1420:GOSUB
 1420:SOUND 0,0,0,0
1350 POKE 656,2:POKE 657,15:X=0:D=6
1360 ? "YRIZ IPIOS"
1370 X=X+1:POKE 755,0:GOSUB 1420
1380 POKE 755,1:GOSUB 1420
1390 IF X<20 THEN 1370
1400 Z=Z-2:IF Z<0 THEN Z=0
1410 GOTO 1280
1420 DE=0
1430 DE=DE+1:IF DE=D THEN RETURN
1440 C=C+16:IF C>255 THEN C=8
1450 IF STRIG(0)=0 OR STRIG(1)=0 OR PE
EK(53279)=6 THEN POKE 559,0:GOTO 1480
1460 POKE 710,C
1470 GOTO 1430
1480 POP :FOR I=53248 TO 53251:POKE I,
0:NEXT I:MOVES=0:? CHR$(125):SOUND 0,0
,0,0:POKE 77,0:GOTO 150
1490 REM BOOBY
1500 BONUS=0:POSITION A,D:? #6;" ":RES
TORE 1510:POKE 1536,0:POKE 704,154
1510 DATA 126,60,24,60,60,60,60,24,90,
126,60,36,153,126,129
1520 FOR I=1571 TO 1578:READ X:POKE I,
X:NEXT I:POKE 53768,0:POKE 53761,168
1530 FOR I=4 TO PD STEP 2:POKE 1536,I:
POKE 53760,I:NEXT I:POKE 1536,0
1540 FOR I=1572 TO 1578:READ X:POKE I,
X:NEXT I:POKE 1536,PD
1550 FOR J=15 TO 0 STEP -1:FOR K=2 TO
10 STEP 2:POKE 704,K*J:SOUND 0,J*K,2,J
:NEXT K:NEXT J:POKE 704,10
1560 FOR DE=1 TO 50:NEXT DE
1570 FOR I=1 TO 2
1580 POSITION 6,9:? #6;"st vmtuo":SOUN
D 0,200,10,10:FOR DE=1 TO 20:NEXT DE
1590 POSITION 6,9:? #6;"          ":SOUN
D 0,0,0,0:FOR DE=1 TO 20:NEXT DE
1600 NEXT I
1610 POSITION 6,9:? #6;"HHHHHHHH"
1620 POKE 53248,0:POKE 704,14
1630 IF RE=0 THEN POSITION A,D:? #6;P$
(GO,GO)
1640 IF RE=1 THEN POSITION A,D:? #6;P$
(3-GO,3-GO)
1650 RESTORE 220:FOR I=1571 TO 1578:RE
AD X:POKE I,X:NEXT I:GOTO 610
1660 REM BONUS
1670 POSITION A,D:? #6;" ":RESTORE 168
0:POKE 53248,0:POKE 704,106
1680 DATA 60,126,90,126,90,66,102,60
1690 FOR I=1571 TO 1578:READ X:POKE I,
X:NEXT I
1700 POKE 53248,PA:BONUS=2
1710 FOR J=15 TO 2 STEP -1:FOR K=15 TO
 1 STEP -3:SOUND 0,J*K,14,15:POKE 704,
J*K:NEXT K:NEXT J:SOUND 0,0,0,0
1720 POKE 704,106
1730 FOR DE=1 TO 50:NEXT DE
1740 FOR I=1 TO 2
1750 SOUND 0,50,10,10:POSITION 7,9:? #
6;"ntwlro":FOR DE=1 TO 20:NEXT DE
1760 SOUND 0,100,10,10:POSITION 7,9:?
#6;"NTWLRO":FOR DE=1 TO 20:NEXT DE
1770 NEXT I
1780 SOUND 0,0,0,0
1790 POSITION 7,9:? #6;"HHHHHH"
1800 POKE 53248,0:POKE 704,14
1810 POSITION A,D:? #6;P$(GO,GO)
1820 RESTORE 220:FOR I=1571 TO 1578:RE
AD X:POKE I,X:NEXT I:GOTO 420
1830 REM MYSTERY SQUARES
1840 IF RE=0 THEN POKE 710,152
1850 IF RE=1 THEN POKE 710,88
1860 FOR I=1 TO MN
1870 A=INT(RND(0)*8)+6
1880 D=INT(RND(0)*7)+2
1890 LOCATE A,D,Z:IF Z<>91 THEN GOTO 1
870
1900 IF A=6 AND D=8 THEN 1870
1910 POSITION A,D:? #6;P$(3+RE,3+RE)
1920 FOR Q=1*15 TO 0 STEP -6:SOUND 0,0
,10,10:NEXT Q:SOUND 0,0,0,0
1930 NEXT I:RETURN
1940 REM DRAW GRID
1950 POKE 1538,131:POKE 1539,131
1960 A=5:X=100
1970 POSITION 8,0:? #6;"PUqn"
1980 POSITION A,1:? #6;"BGGGGGGGGE"
1990 FOR D=2 TO 8
2000 POSITION A,D:? #6;"AIIIIIIIIF"
2010 NEXT D
2020 POSITION A,9:? #6;"CHHHHHHHHD"
2030 POKE 559,46
2040 FOR DE=1 TO 40:NEXT DE
2050 GOTO 880
2060 GOTO 720
2070 FOR DE=1 TO 40:NEXT DE
2080 POKE 53250,66:POKE 53251,158
2090 FOR I=130 TO 103 STEP -1:X=-X:SOU
ND 0,100+X,10,10:POKE 1538,I:POKE 1539
,I:NEXT I:SOUND 0,0,0,0:RETURN
2100 REM DLI
2110 DL=PEEK(560)+PEEK(561)*256:POKE D
L+14,135
2120 RESTORE 2140
2130 FOR I=1771 TO 1790:READ X:POKE I,
X:NEXT I
2140 DATA 72,138,72,169,0,162,0,141,10
,212,141,24,208,142,26,208,104,170,104
,64
2150 POKE 512,235:POKE 513,6:RETURN
2160 REM NEW CHARACTERS
2170 POKE 106,PEEK(106)-8:GRAPHICS 2:P
OKE 710,0:POKE 711,54:POKE 752,1:POKE
656,0:POKE 657,15:? " by MARKY "
2180 POSITION 8,5:? #6;"GRID"
```

GRID CON'T

```
2190 GOSUB 2100
2200 ST=(PEEK(106)+4)*256
2210 IF PEEK(1615)<>104 THEN GOSUB 259
0
2220 FOR I=0 TO 255:POKE I+ST,PEEK(573
44+I):NEXT I
2230 POKE 656,0:POKE 657,13:? "PLEASE
WAIT"
2240 RESTORE 2300
2250 FOR I=256 TO 487:READ C:POKE I+ST
,C:NEXT I
2260 FOR I=488 TO 1000:POKE I+ST,PEEK(
57344+I):NEXT I
2270 GOSUB 3080
2280 POKE 559,0:POKE 756,ST/256:? CHR$
(125)
2290 RETURN
2300 DATA 126,90,126,60,36,153,126,129

2310 DATA 7,7,7,7,7,7,7,7
2320 DATA 0,0,0,0,0,7,7,7
2330 DATA 7,7,7,0,0,0,0,0
2340 DATA 224,224,224,0,0,0,0,0
2350 DATA 0,0,0,0,0,224,224,224
2360 DATA 224,224,224,224,224,224,224,
224
2370 DATA 0,0,0,0,0,255,255,255
2380 DATA 255,255,255,0,0,0,0,0
2390 DATA 0,124,238,238,238,254,238,23
8
2400 DATA 0,60,126,126,126,126,60,0
2410 DATA 126,102,6,30,24,24,0,24
2420 DATA 0,252,238,252,238,238,254,25
2
2430 DATA 0,124,230,224,224,230,254,12
4
2440 DATA 0,248,236,230,230,238,254,25
2
2450 DATA 0,254,224,254,224,224,254,25
4
2460 DATA 0,124,230,224,238,230,254,12
6
2470 DATA 0,60,60,60,60,60,60,60
2480 DATA 0,224,224,224,224,224,254,25
4
2490 DATA 0,238,246,254,254,238,238,23
8
2500 DATA 0,124,238,238,238,238,254,12
4
2510 DATA 0,252,238,252,238,238,238,23
8
2520 DATA 0,124,224,124,14,14,254,252
2530 DATA 0,238,238,238,238,238,254,12
4
2540 DATA 0,230,230,230,218,254,254,23
0
2550 DATA 0,252,230,230,254,252,224,22
4
2560 DATA 0,238,238,124,56,56,56,56
2570 DATA 255,129,129,129,129,129,129,
255
2580 DATA 0,60,60,60,60,0,60,0
2590 REM M/C P/M
2600 RESTORE 2610:FOR I=0 TO 234:READ
J:POKE 1536+I,J:NEXT I:RETURN
2610 DATA 0,0,0,0,0
2620 DATA 0,0,0,0,0
2630 DATA 0,0,0,0,0
2640 DATA 0,0,0,0,0
2650 DATA 0,0,0,0,0
2660 DATA 0,0,35,6,45
2670 DATA 6,55,6,65,6
2680 DATA 0,0,0,0,0
2690 DATA 0,0,0,0,0
2700 DATA 0,0,0,0,0
2710 DATA 0,0,0,0,0
2720 DATA 0,0,0,0,0
2730 DATA 0,0,0,0,0
2740 DATA 0,0,0,0,0
2750 DATA 0,0,0,0,0
2760 DATA 0,0,0,0,104
2770 DATA 162,6,160,90,169
2780 DATA 7,32,92,228,96
2790 DATA 173,26,6,208,6
2800 DATA 32,187,6,32,104
2810 DATA 6,76,98,228,169
2820 DATA 3,133,203,166,203
2830 DATA 189,0,6,133,205
2840 DATA 188,8,6,157,8
2850 DATA 6,189,75,6,133
2860 DATA 204,138,10,170,189
2870 DATA 16,6,133,206,189
2880 DATA 17,6,133,207,189
2890 DATA 27,6,133,208,189
2900 DATA 28,6,133,209,166
2910 DATA 204,169,0,145,206
2920 DATA 136,192,255,240,3
2930 DATA 202,16,246,164,204
2940 DATA 177,208,164,205,145
2950 DATA 206,136,132,205,192
2960 DATA 255,240,4,198,204
2970 DATA 16,237,198,203,16
2980 DATA 178,96,162,3,173
2990 DATA 24,6,133,206,173
3000 DATA 25,6,133,207,188
3010 DATA 12,6,189,231,6
3020 DATA 49,206,145,206,188
3030 DATA 4,6,189,227,6
3040 DATA 17,206,145,206,152
3050 DATA 157,12,6,202,16
3060 DATA 229,96,2,12,32
3070 DATA 192,252,242,207,47
3080 REM *** P/M INITIALIZATION ***
3090 I=PEEK(106)-8:POKE 54279,I:POKE 5
3277,3:PMBASE=I*256
3100 MISSILE=PMBASE+384:J=INT(MISSILE/
256):POKE 1560,(MISSILE-J*256):POKE 15
61,J
3110 PLAYER0=PMBASE+512:J=INT(PLAYER0/
256):POKE 1552,(PLAYER0-J*256):POKE 15
53,J
3120 PLAYER1=PMBASE+640:J=INT(PLAYER1/
256):POKE 1554,(PLAYER1-J*256):POKE 15
55,J
3130 PLAYER2=PMBASE+768:J=INT(PLAYER2/
256):POKE 1556,(PLAYER2-J*256):POKE 15
57,J
3140 PLAYER3=PMBASE+896:J=INT(PLAYER3/
256):POKE 1558,(PLAYER3-J*256):POKE 15
59,J
3150 FOR J=PMBASE TO PMBASE+1024:POKE
J,0:NEXT J:A=USR(1615):RETURN
```

MEETING

WED. JUNE 11TH

SOUTH EUGENE

HIGH

ADVENTURES WITH THE MANDELBROT SET

USING AN ATARI 800

or a slow road to insanity.... by Rita Plukss, Pres. M.A.C.E.
(reprinted from Melbourne ACE, POB 340, Rosanna, Victoria, Australia 3084 Mar 86)

Girolama Cardano was an astrologer and physician to kings, a compulsive gambler who spent his life on the verge of bankruptcy and prison as well as on the brink of atheism and heresy. He was also a scientific writer from whose quill books flowed in fountains.

Over 300 years later he has been responsible for my near compulsive behaviour and for grave doubts being cast upon my sanity. Because of his astounding concept of negative numbers, and fictitious or sophisticated quantities, my family have placed a sign on the computer room door — quarantine area — and refuse to come in close contact with me. All they hear through the crack in the door, or when I emerge for sustenance is "why is it not there?", "where is it?". They just shake their heads and hope that there is a power failure and then I will return to human form and remember all those tasks that now remain undone.

What Cardano put forward is today called an "imaginary" number. This is the square root of a negative number. What? you say. Has she finally flipped her lid? We all agree that it is impossible to have a square root of a negative number, but, just suppose, if it could be imagined and we combined this imaginary number with a real number eg 1+2/-2, then, the result would be known as a "complex" number.

In fact these "complex" numbers have all sorts of applications in areas such as atomic physics, engineering etc. But who can understand such highly abstracted concepts? All I'm interested in is using them to generate pretty pictures.

A "complex" number is one that is composed of an ordinary number plus some multiple of the imaginary unit (the square root of minus one). eg 2+5/-1. The customary way of writing /-1 in such numbers is i, and any complex number as a+bi. In our example a is 2, b is 5 and i is the square root of minus one. So we can write it as 2+5i.

Ordinary numbers can all be thought of as lying along a single straight line without any gaps in it - a continuum. But a typical complex number a+bi has no place on the line of ordinary numbers. So what can you do with it? Where does it fit?

Gauss proposed that the complex number could be thought of as labelling a point on a two dimensional plane; where a would be the horizontal distance and b the vertical distance and that the full a+bi could determine the position of a point on the plane just as the x and y of a Cartesian coordinate couplet determines the point on a graph.

So where does all this lead to? It leads to the Mandelbrot Set. The Mandelbrot set is named for Benoit B. Mandelbrot, a research fellow at the IBM Thomas J. Watson Research Centre in Yorktown Heights New York. He developed the field of fractal geometry (the mathematical study of forms having a fractional dimension). This work was carried on by John H. Hubbard, a mathematician at Cornell University and Hubbard was one of the first people to make computer generated images of the Mandelbrot set. According to Hubbard it is "the most complicated object in mathematics".

If you wish to gain a more mathematical insight into the Mandelbrot set, then you will need to have some understanding of two mathematical concepts - iterative procedures and complex numbers. The first is fairly easy (just like FOR-NEXT loops) but the second is a bit more abstruse. The articles on the Mandelbrot set in both Scientific American (August 1985) and Your Computer (January 1986) will give you sufficient background of both concepts, but better still, consult with a senior high school maths student.

To reiterate, a complex number takes the form of C=A+Bi where both A and B are real numbers (eg -5, 6, 8.996 etc) and i is defined as the square root of -1. The complex number C consists of a real part (A) and an imaginary part (Bi). We can now describe the iterative procedure that generates the Mandelbrot set. Start with the expression Z=Z*Z+C where both Z and C are complex numbers and allow Z to vary. Set Z=0 to give us Z=0+C=C (C is now equal to Z); substituting this value back into the equation for the next iteration will give Z=C*C+C, then Z=(C*C+C)(C*C+C)+C and so on. The value of Z fluctuates with successive iterations because we have introduced the value i*i=-1 into our calculations. The Mandelbrot set is the set of all complex numbers for which the size of Z*Z+C remains finite even after an indefinitely large number of iterations and is situated at the centre of a vast two dimensional plane. The boundary of the set is a fractal (and you thought I'd given up on fractals)! But what a fractal! It must be the ultimate.

Our interest lies on these edges of the set where Z falls outside the Mandelbrot set i.e. where Z does reach the value of 2 and will go off to infinity. The fractal area!

The number 2 is the crucial factor in working with the Mandelbrot set. "A straight forward result in the theory of complex number iterations guarantees that the iterations will drive Z to infinity IF AND ONLY IF at some stage Z reaches the value of 2 or greater." But, actually there are very few points where the value of 2 will not be reached after a small number of iterations. This situation becomes rarer as the iteration count increases.

The Mandelbrot set itself is the black (static) area within the plane, where Z does not reach the value of 2. (That is, black on the screen, white on the printouts.) It is shaped like a squat, wart covered figure eight lying on its side (see Fig.1).

The coloured and patterned areas around the set are where Z has reached the value of 2 and the colour of the pixel should tell you how many iterations it took to reach that value. This is the area that is of greatest interest, the areas that surround the set. These areas are a myriad of colours and patterns, no two areas being identical, yet repeating themselves over and over again. There are riots of organic looking tendrils and circular sweeps, whirls and whorls, and the colour! You can look at the full set, look at the edges, or you can zoom in as with a microscope and go deeper and deeper, the patterns and colours just keep happening.

The program that searches for these values has been called MANDELZOOM (Program listing 1). As the name implies it investigates the Mandelbrot set and also allows you to zoom into the set for closer investigation of any part.

With the Atari 800 we do have a couple of minor problems; no double precision therefore we have to limit our zooming, and the resolution we can actually achieve, once again limiting our final screen output. But with these limitations recognised we can produce acceptable displays within those constraints.

Our first programs produced immediate displays - if anyone is interested in these let me know and I will pass them on to you. The final versions (we think they are the final versions - but we have thought that before!) of the programs that follow this article were put together by Dick. But there have been a small core of enthused people (or nuts?) working diligently through the various stages to come up with the final version. This subset of the Mandelbrot set have worked and bolstered each other and they are Chris Ryan, Dick Kellett, Ron Collis, and of course, myself. There have been a couple of enthusiastic late comers, Gary Fyfe and Ian Conner (both from Geelong). The main problem with immediate screen plotting was that we could not reallocate where the colour changes should take place. Not having the memory available to read the data into an array, Dick had a stroke of genius - put it to disk as a data file, then when you are ready to plot, read it back from the disk! Brilliant I said! Only one problem, each data file is 246 sectors long! Oh well, what else are disks for?

It still amazes me that with only 48K of memory (less BASIC of course), an 8 bit ATARI computer and a TV screen that we could produce screens that are quite acceptable considering that what you see elsewhere have been produced on computers much more powerful than ours.

BRIEF EXPLANATION OF THE 4 PROGRAMS. See Dick's article on how each of the programs work.

PROGRAM 1: MANDELZOOM - MANDLCAL

Performs the calculations on that part of the Mandelbrot set under investigation and writes it as a data file to disk (15-30 hours), depending on how much of the area is within the actual set (black).

PROGRAM 2: MANDPLOT

Reads the data file obtained by program 1 and plots that data to screen. (15-20 minutes)

PROGRAM 3: DATACHK

Scans the data disk, reads the number of iterations required to generate each piece of data (pixel) and collates this information. This allows you to see at what levels of iteration to make your colour changes when running program 2.

PROGRAM 4: COLDUMP

This has nothing to do with the Mandelbrot set, but is a screen dump to dump each colour register separately to the printer (7+ screen). By using this, colour carbon paper and a good eye when you roll your paper back, you too can produce colour printouts as were seen at the February meeting.

Have fun and good luck! For more information on the Mandelbrot set refer to Scientific American (August 1985) or if really involved try The Fractal Geometry of Nature by Benoit B.Mandelbrot (W.H.Freeman & Co. New York, 1977) and Fractals - Form, Chance and Dimension by Benoit B.Mandelbrot (W.H.Freeman & Co. San Francisco, 1977).

(Rita and Dick want to trade programs and information about this facinating new subject with you — ed.)

FLOATING POINT

XE (Reprint: CompUtah, April 1986)

The Atari has built-in ROM routines to take care of most of the floating point (FP) needs to easily enable a number of functions which the 6502 microprocessor does not readily offer in its instruction codes. Basic makes calls to these same ROM routines and they are available to the assembly language programmer, too. The functions offered are:

1. ASCII to FP and FP to ASCII conversion.
2. Integer to FP and FP to integer conversion.
3. FP add, subtract, multiply and divide.
4. FP log, exponentiation and ploynomial evaluation.
5. FP number clear, load, store and move.

The FP package uses page zero variables from $D4 to $FF and a buffer area from $57E to $5FF. If your routine is not making any calls to any of the FP package, these areas are available for your use. Keep in mind that Basic makes calls to the FP package so these areas are "sacred" if you are using Basic.

We will first equate and define our floating point variables, and then will equate the routine call addresses.

```
;;Floating Point Variable Equates
;==============================
FR0 = $D4 ;6 byte floating point register
FR1 = $E0 ;6 byte floating point register
FLPTR = $FC ;2 byte pointer to a FP number
INBUFF= $F3 ;2 byte pointer to an ASCII text buffer
CIX = $F2 ;1 byte index to buffer pointed in INBUFF
LBUFF = $580 ;result buffer for conversions to ASCII
;;Floating Point Routine Addresses
;==============================
AFP = $D800 ;ASCII to FP
FASC = $D8E6 ;FP to ASCII
IFP = $D9AA ;Integer to FP
FPI = $D9D2 ;FP to Integer
FADD = $DA66 ;Add 2 FP #'s & check if out of range
FSUB = $DA60 ;Subtract 2 FP #'s & check if out of range
FMUL = $DADB ;Multiply 2 FP #'s & check if out of range
FDIV = $DB28 ;Divide 2 FP #'s & check if out of range
LOG = $DECD ;natural logarithm
LOG10 = $DED1 ;base 10 logarithm
EXP = $DDC0 ;exponentiation base e
EXP10 = $DDCC ;exponentiation base 10
PLYEVL= $DD40 ;FP polynomial evaluation
ZFR0 = $DA44 ;zero out FR0
ZF1 = $DA46 ;zero out FR1
FLD0R = $DD89 ;load register FR0 from X,Y pointer to FP
FLD0P = $DD8D ;load register FR0 from FLPTR pointer to FP
FLD1R = $DD98 ;load register FR1 from X,Y pointer to FP
FLD1P = $DD9C ;load register FR1 from FLPTR pointer to FP
FST0R = $DDA7 ;store from FR0 by X,Y pointer to dest FP
FST0P = $DDAB ;store from FR0 by FLPTR to dest FP
FMOVE = $DDB6 ;move FP from FR0 to FR1
```

FUNCTION CALLING SEQUENCES

The following will describe how to call some of the routines. Other routines are and their action is clear from their descriptions above. The more complex ones including exponentiation, logarithms, and polynomial evaluation are described in the Atari technical notes.

ASCII to FP (AFP)

In:
INBUFF points to buffer with ASCII string.
CIX is offset to 1st byte of ASCII string.
After string place any non-numeric ASCII value.
Out:
Carry set if 1st ASCII number invalid.
FR0 = FP #
CIX = buffer offset to 1st byte after ASCII string.
Routine scans string until it reaches invalid byte.

FP to ASCII (FASC):

In:
FR0 = FP
Out:
INBUFF = pointer to 1st byte of ASCII string, last byte will have bit 7 set.

Integer to FP (IFP)

In:
FR0 = LSB of integer
FR0+1 = MSB of integer
Out:
FR0 = FP of integer

FP to Integer (FPI)

In:
FR0 = FP
Out:
Carry set if integer greater than 65535 or if FP is negative
FR0 = LSB of integer
FR0+1 = MSB of integer
Routine does true rounding in conversion.

FP Add (FADD)

In:
FR0 = FP
FR1 = FP
Out:
Carry set if out of range
FR0 = sum FR0+FR1
FR1 is altered

FP Subtract (FSUB)

In:
FR0 = minuend
FR1 = subtrahend
Out:
Carry set if out of range
FR0 = result FR0-FR1
FR1 is altered

FP Multiply (FMUL)

In:
FR0 = multiplier
FR1 = multiplicand
Out:
Carry set if out of range
FR0 = product FR0*FR1
FR1 is altered

FP Divide (FDIV)

In:
FR0 = dividend
FR1 = divisor
Out:
Carry set if out of range or divide by zero
FR0 = quotient of FR0/FR1
FR1 is altered

PC REPAIR

Reprint: Foghorn, May 1986

. . . Removing bad chips from a board can be difficult, but even more frustrating can be the problems you experience when you solder in a new part — seemingly making a perfect soldering job — and it won't work afterwards.

Very often these days, the more complicated circuit boards are "multi-layer". They not only have conductors on the top and bottom of the board, where you can see the quality of your soldered connection, but also have more conductors inside the circuit board where you can see nothing, but suspect a lot!

About 20 years ago a repairman taught me a tip which I have carried with me into the modern age. Those were the days when discrete components were soldered to metal lugs. Very often the factory wrapped the wires round the lug before soldering, creating something almost impossible to undo. The tip I was given was to not even try; just clip off the wires close to the body of the discrete resistor, capacitor or whatever, and use the wire stumps as new terminals around which to wrap the wires of the replacement part.

I find it possible to use this tip in replacing integrated circuits as well as other modern parts. Just clip off the legs with a small, sharp cutter about halfway between the chip body and the circuit board. Use the cutter to slightly shorten the legs of the replacement chip, and sit the chip slightly offset from the old position so its legs lie against the leg stumps of the old chip. Now solder them one at a time to the stumps and you're done.

It won't look beautiful, but this repair is faster because you never unsolder the old chip. This method is usually more reliable because you don't disturb the factory soldering job between the old legs and the multiple layers of the circuit board. Nor are you putting mechanical stress on the board which might crack it, as sometimes happens while pulling off old chips.

In many cases there is enough clearance so the new chip can sit a little high compared to the others. This makes installation even easier. The only places this repair technique might fail is where the circuits operate at high enough frequencies to make the design sensitive to the chip leg dress, or where the quarters are so cramped as to leave no working space in the chip side of the board.

— Alan Chattaway

ST KEYBOARD

I have built a detached keyboard for the ST! It uses the original ST keyboard, original keyboard top cover (bearing the Atari 520 ST logo), and a few (new, but surplus) parts (see below).

Hardly any electronics experience is needed for this project. Most of it depends on how well you can construct an enclosure for the keyboard. To me, the whole thing looks much classier: The plexiglass I bought is dark satin brown, and it contrasts perfectly with the ST's native color. So far, it has been in daily use for about 5 weeks without any problems. The coiled cord from the keyboard plugs into the "main" box, the mouse (and the joystick, when we have some need for it) plugs into the keyboard housing. If need be, it can be reassembled into the way it originally was, except for the saw cut made along the deep groove you find running along the entire width above the function keys and the 520 ST logo.

Connections: The keyboard plugs into the main board using an 18-pin connector (J12). These pins are numbered 1 through 18, one starting at the bottom (closest to the keyboard). Only five of these need go to the board; of the remaining 13 pins, one is a keyway ("pin" 2 which is absent), pin 18 is unconnected, and eleven send/receive signals from the mouse/joystick0 and joystick1 ports:

Mouse J12	Joystick1	J12
4rt/YB 4	8	left 3
3 - 5	unconnected	right 4
5	179	reservd
		unconnected

The main reasons I moved the DB-9s to the keyboard were: (1) I could not find a coiled cord with 16 conductors (I cannot stand external ribboncables); and, (2) I feel the mouse and keyboard should be close. Alas, a typical phone extension chord has only 4 conductors, and we need five:

J12-16	
15	¾ reset (generated on the main board)
14	Tx Data (of U21 6850 ACIA)
13	Rx Data (of U21 6850 ACIA)
	+5V
	1 ground

Parts Needed: Coiled cord of sufficient length; min. 5 conductors; 5-, or 6-pin DIN socket and plug; two DB-9s or one 18-pin plug; 0.1" pin separation (can use a plug with more than 18 pins); one 18-pin socket (can use up to 20-pin socket); and some scrap plexiglass sheets.

Construction: Most of it is this "box" making with a few solder connections as indicated above. To instruct how to do the construction is so tedious that I am going to suggest the very obvious: Open up your ST, and carefully examine it until you discover your own way of doing it. Two cautionary remarks, however: The plexiglass enclosure should be constructed with precision. My wood-working imprecision made me ruin a couple of panels before I got everything accurate to within a millimeter! And the edges should be real straight. Underneath the keyboard are a few chips; make sure you allow for them and do not make the keyboard rest physically on these. Have fun!

— prabhaker matetidep! pm department of computer engineering & science pm, Case Western Reserve University, Cleveland, OH 44106

DESIGN SUGGESTION

XE (Reprint: Puget Sound Atari News)

Are you a hardware hacker, looking for some place to add a couple of chips to your 8 bit computer? But you don't want to add something which will make it incompatible with some future peripheral?

Well, I have a suggestion for you. I have been in the process of adding memory, a real time clock, and several other items to my machine. These additions require several chips to be mapped into the computer. But finding a place to map the chip which will not interfere with some future device is difficult to find. My suggestion is to re-map one of the existing chips opening up a new open area for your own modifications.

Many chips in the computer are selected using the minimum circuitry required. For this reason, there is much unused memory in the $D000 to $D7FF area. An example is the PIA chip. A quick examination of Mapping the Atari Revised, by Ian Chadwick will reveal that the PIA chip occupies the area from $D300 to $D3FF. It also says that locations of $D304 to $D3FF are repeats of locations $D300 to $D303. To operate the PIA only 4 locations are used. This leaves 252 locations potentially open to other uses in that area alone.

Now, why re-map the chip, when there appears to be a couple of areas presently available? Well, another look at the Mapping the Atari Revised will show $D100-$D1FF is unused, as well as $D600-$D7FF. Mightn't it be easier to just use one of these other areas to add something? Then you will not have to modify the computer. Well, it might be. Now, the problem occurs if you want to attempt to remain compatible with any future peripherals. Hardware built to use the special features of the parallel bus will be already in the $D100-$D1FF region, so this is out. The $D600-$D7FF region is presently unused, but I expect commercial hardware designers will realize this area can be accessed from the parallel bus with no modifications to the computer. This area is very likely to be exploited in the future.

In re-mapping a chip, you are making a hardware modification which is not able to be simulated from the outside of the computer. The mod should be save from software since the software is written using the stated addresses for the devices and not the repeat locations.

Ok, now which chip can I re-map? Are there any easy chips? Yes, I used the PIA example above since this is the one with which I am most familiar. I choose the PIA since there are two chip select lines on the chip which are not used. This means I can use some other circuit to disallow the chip from being accessed whenever the area $D304-$D3FF. In my modification I want to maintain the minimum select circuitry, so for reasons of simplicity, I break the area in half so I can have 128 bytes to play with.

This mod only requires me to take the A7 line and run it through an inverter and connect it to the unused chip select line(s). My other chips can then be added by using the PIA chip select and combining this with the A0-A7 address lines to determine which is selected. In this case, if the A7 line is low the PIA chip is selected (like normal). If the A7 line is high, then my modification is selected.

— Bill Penner (KCACES)

DANGER

XE (Reprint: Phoenix, April, 1986)

This technical note is being written because so many of the memory expansion schemes I see being touted are NOT compatible with a standard 130XE. If you implement the memory expansion per most of these schemes, you will be missing one important feature of the 130XE: The ability to direct ANTIC to do its DMA to either main memory or the requested bank of memory. In a standard 130XE, clearing bit 5 to zero requests ANTIC to follow the bank switching; setting bit 5 to a one tells it to remain in main memory, no matter what memory bank was requested.

This is an important feature! Mark Rose (also of OSS) and I will take credit for being instrumental in the creation of the function of this bit. When Atari asked us to do DOS 2.5 and its RamDisk, their prototype hardware had ANTIC following bit 4 along with the CPU. The most obvious problem with this is you can't use the extra banks for CPU purposes (e.g., RamDisk) when ANTIC is doing its DMA in the memory between $4000 and $7FFF. The problem was especially acute with AtariWriter (the 16k cartridge version), since its display memory is ALWAYS in this range.

Actually, Mark and I found if you are ONLY using the bank select memory for a RamDisk, this si not an onerous restriction. It simply means pseudo-sector transfers only occur during vertical blank. In fact, DOS 2.5 still has a flag in it which you can POKE which will tell it to only use extended memory during deferred vertical blank.

Now, there is another hardware solution which we mentioned to Atari: Simply never allow ANTIC to use extended memory. We discussed the two options with Atari, and both they and we decided we felt strongly that the capability of bank selecting ANTIC's memory was important. Thus the use of that bit.

So, if yuour 800XL hardware mod works with the AtariWriter cartridge, then you obviously adopted that second hardware solution: Don't let ANTIC use extended memory. That is not a really terrible decision (especially if it is economically motivated), but it does mean that it is possible that some future 130XE software will not run on your modified machine. I already have at least one piece of software, written in ACTION!, which depends on the 130XE's method. But it's only an ultra-fast picture switching demo, so it's no big deal.

There is a mod to both the 130XE and 800XL which maintains the 130XE/ANTIC bank select capability. It is designed by [ACE member] Charles Andrews of Eugene, Oregon. He showed a 320XE using this mod at CES in January (in Atari's booth, as a courtesy to him by Atari — though it does appear to be an implicit endorsement of his scheme). I believe his method uses an entirely separate port for controlling the beyond-130XE extensions (in the $D6xx range, maybe?). However, I devised a method of doing the same thing using only Port B. The scheme is outlined in the following paragraph.

A "legal" 320XE: This mod depends on the fact that the diagnostic ROM area is only used at power up or by the self-test routines. At these times, both ANTIC and the CPU are using only main memory, so bits 4 and 5 of $D301 are both set to one. Thus we change the "enable" of the diagnostics from the logic equation diag_enable = not_bit7 and bit4 and bit5.

Then the enable for the extended RAM becomes RAM_enable = (not_bit4 or not_bit5) and we can now use bits 6 and 7 for bank selection in the same manner other schemes use bits 6 and 5. The reason this works: Even if Atari ever changes the self-diagnostics so they check the extended RAM, they can't put that particular code in the ROM which overlays $5000-$57FF, because that's right in the middle of the RAM area they need to check!

— Bill Wilkinson

Atari Computer Enthusiasts

The ACE Newsletter is the first Atari user group newsletter to be published anywhere in the world. We began publishing in April, 1981.

A.C.E. is an independent, non-profit and tax exempt computer club and user group with no connection to Atari Corp. We are a group interested in educating members in the use of Atari computers and publishing the latest News, Reviews and Rumors.

All our articles, reviews and programs come from you, our members.

Our membership is world wide. Membership fees include the A.C.E. newsletter (10 issues per year) and high access in the A.C.E. Bulletin Board System. Dues are $14 a year for U.S., and $24 a year overseas airmail.

Subscription Dept.: 3662 Vine Maple Dr., Eugene, OR 97405

President — Dick Barkley, 2907 Wingate, Eugene, OR 97405 (503) 344-5843

Vice President — Larry Gold, 1927 McLean Blvd., Eugene, OR 97405 (503) 686-1490

8-Bit Librarian — Chuck & Jody Ross, 2222 Ironwood, Eugene, OR 97401 (503) 343-5545

ST Librarian — Jim Bumpas, 4405 Dillard Road, Eugene, OR 97405 (503) 484-4746

Editors — Mike Dunn, 3662 Vine Maple Dr., Eugene, OR 97405 (503) 344-6193

Jim Bumpas, 4405 Dillard Road, Eugene, OR 97405 (503) 484-4746

Larry Gold, 1927 McLean Blvd., Eugene, OR 97405 (503) 686-1490

E.R.A.C.E (Education SIG Editor) — Nora Young, 105 Hansen Lane Eugene, OR 97404 / (503) 688-1458

Send 50c stamps or coin ($1 overseas) to the Librarian for updated ACE Library lists.

A.C.E. Bulletin Board
(503) 343-4352

On line 24 hours a day, except for servicing and updating. Consists of a 130 XE upgraded to 320k by Charlie Andrews, 2 double-density disk drives, and a 10-meg hard disk from Supra Corp, an Epson MX80 printer, a 2400 US Robotics Courier modem, running the PD/M Bulletin Board software.

Best of ACE books

Volume 1 contains bound issues of the ACE Newsletter from the first issue, Oct. 1981 to June, 1982.

Volume 2 contains July 1982 through June 1983.

Only $12 each ($2 extra for airmail). Available only from George Suetsugu, 45-602 Apuapu St., Kanoehe, HI 96744.

TYPESETTING
FROM YOUR COMPUTER

ATARI OWNERS: If you have a modem, text editor, and communications program to send ASCII files, you should consider the improved readability and cost savings provided by **TYPESETTING** your program documentation, manuscript, newsletter, or other lengthy text instead of just reproducing it from line printer or daisy-wheel output. Computer typesetting by telephone offers you high quality, space-saving copy that creates the professional image you want! Hundreds of type styles to choose from with 8 styles and 12 sizes "on line." And it's easy to encode your copy with the few typesetting commands you need.

COMPLETE CONFIDENTIALITY GUARANTEED
— Bonded for your protection —
PUBLICATION DESIGN, EDITING, & PRODUCTION

Editing & Design Services

30 East 13th Avenue Eugene, Oregon 97401
Phone 503/683-2657

SortFinder Index

A composite index of Atari related articles in four popular computer periodicals, including the ACE Newsletter. Volume 1 covers April, 1981 through June, 1983. Volume 2 covers July, 1983 through December, 1985. Only $6 per printed copy or $11 per disk copy for ACE members. Available from: Jim Carr, 2660 S.W. DeArmond, Corvallis, OR 97333.

ATARI
COMPUTER
ENTHUSIASTS
3662 Vine Maple Dr. Eugene OR 97405

FIRST CLASS MAIL